

Real-Time Monitoring of Data Streams using Machine Learning Techniques for Optimizing Decision Making



Kyriakos Skarlatos

Supervisor: Prof. Sotiris Bersimis

Department of Business Administration
University of Piraeus

This dissertation is submitted for the degree of
Doctor of Philosophy

Παράρτημα Β: Βεβαίωση Εκπόνησης Διδακτορικής Διατριβής



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ ΣΧΟΛΗ ΟΙΚΟΝΟΜΙΚΩΝ ΕΠΙΧΕΙΡΗΜΑΤΙΚΩΝ ΚΑΙ ΔΙΕΘΝΩΝ ΣΠΟΥΔΩΝ ΤΜΗΜΑ ΟΡΓΑΝΩΣΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ ΕΠΙΧΕΙΡΗΣΕΩΝ

ΒΕΒΑΙΩΣΗ ΕΚΠΟΝΗΣΗΣ ΔΙΔΑΚΤΟΡΙΚΗΣ ΔΙΑΤΡΙΒΗΣ

(περιλαμβάνεται ως ξεχωριστή (δεύτερη) σελίδα στο σώμα της διδακτορικής διατριβής)

Δηλώνω υπεύθυνα ότι η διδακτορική διατριβή για τη λήψη του διδακτορικού τίτλου, του Τμήματος Οργάνωσης και Διοίκησης Επιχειρήσεων του Πανεπιστημίου Πειραιώς, με τίτλο: *Real-Time Monitoring of Data Streams using Machine Learning Techniques for Optimizing Decision Making* έχει συγγραφεί από εμένα αποκλειστικά και στο σύνολό της. Δεν έχει υποβληθεί ούτε έχει εγκριθεί στο πλαίσιο κάποιου άλλου διδακτορικού, μεταπτυχιακού προγράμματος ή προπτυχιακού τίτλου σπουδών, στην Ελλάδα ή στο εξωτερικό, ούτε είναι εργασία ή τμήμα εργασίας ακαδημαϊκού ή επαγγελματικού χαρακτήρα.

Δηλώνω επίσης υπεύθυνα ότι οι πηγές στις οποίες ανέτρεξα για την εκπόνηση της συγκεκριμένης εργασίας, αναφέρονται στο σύνολό τους, κάνοντας πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του διπλώματός μου.

Υπογραφή Διδάκτορα.....

Ονοματεπώνυμο Κυριάκος Σκαρλάτος

Ημερομηνία 11/03/2026

I would like to dedicate this thesis to my loving foster parents . . .

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Kyriakos Skarlatos

March 2026

Acknowledgements

I would like to express my deepest gratitude to all those who have supported me throughout the course of my research and the writing of this thesis. First and foremost, I would like to thank my supervisor, Sotiris Bersimis, for his invaluable guidance, expertise, and unwavering support. Their insightful feedback and encouragement have been instrumental in shaping this work. I would also like to extend my sincere thanks to the members of my thesis committee, Polychronis Economou and Petros Maravelakis, for their thoughtful suggestions and constructive criticism.

My heartfelt thanks to my foster family (Grammatikopoulos), whose quiet and steadfast support during my early years laid the foundation for all I have achieved. To my foster parents, Stella & Christo, I am forever grateful for the stability, care, and compassion you provided during those formative times. Although your help may not have been directly visible in my academic journey, it was in the nurturing environment you created that I found the strength and resilience to pursue my goals. Your unwavering belief in me, even when I was too young to fully understand it, has shaped the person I am today. I would also like to thank my foster siblings, Elias and Tasos, whose companionship, understanding, and shared experiences enriched my life and made my journey more meaningful.

A special thanks go to my friends, whose constant love, patience, and understanding have provided me with the strength and motivation to complete this project. Finally, I would like to thank all those who, in one way or another, contributed to the successful completion of this thesis. Although they are too numerous to mention individually, their help has been invaluable.

Abstract

The growth of big data is primarily driven by the increasing digitization of information and the widespread use of data-collecting devices. As technological advancements continue, the volume, variety, and velocity of data generation expand, giving rise to big data analytics aimed at extracting valuable insights from vast datasets. The growing volume of data presents both opportunities and significant challenges. In the realm of Statistical Process Monitoring (SPM), the analysis of high-dimensional data often leads to the “curse of dimensionality”, where data sparsity hinders the detection of meaningful patterns and anomalies. Additionally, monitoring complex relationships among multiple variables requires more advanced methods than traditional univariate approaches. Multivariate Statistical Process Monitoring (MSPM) addresses this need by employing tools such as multivariate control charts, notably the Hotelling’s T^2 chart, to capture the joint behavior of correlated quality variables. However, implementing these control charts in real-time, high-dimensional settings is particularly difficult due to the computational demands and the need for rapid decision-making when processing large streams of data. A potential solution is to combine traditional MSPM techniques with modern machine learning approaches; even so, this integration poses challenges related to model interpretability, feature selection, and result integration. In this study, a modern and robust method inspired by cluster validity indexing techniques is presented. This method is compared to the traditional multivariate control charts based on the Hotelling T^2 statistic, as well as to other metrics from the cluster analysis framework, such as the Dunn, Silhouette, Calinski-Harabasz, and Davies-Bouldin indices. Extensive simulation studies demonstrate that the proposed method outperforms existing approaches, particularly in scenarios such as mean drifting and density-related changes, involving data streams with correlated features. At the conclusion, various real-world application scenarios utilizing statistical and machine learning techniques are presented. Ranging from simulations to practical applications, this dissertation seeks to bridge the concepts of statistics and machine learning, with the latter inheriting tools and methodologies from the former.

Περίληψη

Η ραγδαία ανάπτυξη των μεγάλων δεδομένων (big data) οφείλεται πρωτίστως στην εντεινόμενη ψηφιοποίηση της πληροφορίας και στη διευρυμένη χρήση συσκευών συλλογής δεδομένων. Καθώς οι τεχνολογικές εξελίξεις επιταχύνονται, ο όγκος, η ποικιλομορφία και η ταχύτητα παραγωγής δεδομένων αυξάνονται, διαμορφώνοντας το πεδίο της ανάλυσης μεγάλων δεδομένων (big data analytics), το οποίο στοχεύει στην εξαγωγή ουσιαστικής πληροφορίας από εκτεταμένα σύνολα δεδομένων. Ο ολοένα αυξανόμενος όγκος δεδομένων συνεπάγεται σημαντικές ευκαιρίες, αλλά ταυτόχρονα εισάγει και ουσιώδεις μεθοδολογικές και υπολογιστικές προκλήσεις. Στο πλαίσιο της Στατιστικής Παρακολούθησης Διεργασιών (Statistical Process Monitoring -- SPM), η ανάλυση δεδομένων υψηλής διαστατικότητας συχνά οδηγεί στο φαινόμενο της «κατάρας της διαστατικότητας» (curse of dimensionality), όπου η αραιότητα των δεδομένων (data sparsity) δυσχεραίνει την ανίχνευση ουσιαστικών προτύπων και ανωμαλιών. Επιπροσθέτως, η παρακολούθηση πολύπλοκων αλληλεξαρτήσεων μεταξύ πολλών μεταβλητών καθιστά ανεπαρκείς τις παραδοσιακές μονομεταβλητές προσεγγίσεις και επιβάλλει την υιοθέτηση προηγμένων μεθόδων. Η Πολυμεταβλητή Στατιστική Παρακολούθηση Διεργασιών (Multivariate Statistical Process Monitoring -- MSPM) ανταποκρίνεται σε αυτή την ανάγκη μέσω της χρήσης εργαλείων όπως τα πολυμεταβλητά διαγράμματα ελέγχου και ειδικότερα το διάγραμμα T^2 του Hotelling, το οποίο αποτυπώνει τη συλλογική συμπεριφορά συσχετισμένων μεταβλητών ποιότητας. Η εφαρμογή τέτοιων διαγραμμάτων σε περιβάλλοντα πραγματικού χρόνου και υψηλής διαστατικότητας συνοδεύεται από ιδιαίτερες δυσκολίες, οι οποίες απορρέουν από τις αυξημένες υπολογιστικές απαιτήσεις και την ανάγκη ταχείας λήψης αποφάσεων (decision-making) κατά την επεξεργασία εκτεταμένων ροών δεδομένων (data streams). Μια δυνητική κατεύθυνση αντιμετώπισης των δυσκολιών αυτών είναι ο συνδυασμός των παραδοσιακών τεχνικών MSPM με σύγχρονες μεθόδους μηχανικής μάθησης, με στόχο την ενίσχυση της προσαρμοστικότητας και της ανιχνευτικής ικανότητας των συστημάτων παρακολούθησης διεργασιών. Ωστόσο, η ενοποίηση των δύο πλαισίων εγείρει κρίσιμα ζητήματα, όπως η ερμηνευσιμότητα των υποδειγμάτων, η επιλογή κατάλληλων χαρακτηριστικών και η συνεκτική ενσωμάτωση των αποτελεσμάτων σε διαδικασίες λήψης αποφάσεων. Στην παρούσα διατριβή προτείνεται μια σύγχρονη και ανθεκτική μέθοδος πολυμεταβλητής παρακολούθησης διεργασιών, η οποία εμπνέεται από τεχνικές δεικτών εγκυρότητας συστάδων (cluster validity indexing techniques). Η απόδοση της προτεινόμενης μεθόδου αξιολογείται και συγκρίνεται τόσο με τα κλασικά πολυμεταβλητά διαγράμματα ελέγχου που βασίζονται στη στατιστική T^2 του Hotelling, όσο και με μία σειρά δεικτών που προέρχονται από το πεδίο της ανάλυσης συστάδων, όπως οι δείκτες Dunn, Silhouette, Calinski-Harabasz και Davies-Bouldin.

Εκτενείς μελέτες προσομοίωσης καταδεικνύουν ότι η προτεινόμενη προσέγγιση υπερτερεί των υφιστάμενων μεθόδων, ιδίως σε σενάρια ολίσθησης του μέσου (mean drifting) και μεταβολών που σχετίζονται με την πυκνότητα, όταν εξετάζονται ροές δεδομένων με ισχυρά συσχετισμένα χαρακτηριστικά. Τέλος, παρουσιάζονται ενδεικτικές εφαρμογές σε πραγματικά προβλήματα, στις οποίες αξιοποιούνται συνδυαστικά στατιστικές τεχνικές και τεχνικές μηχανικής μάθησης, αναδεικνύοντας τη χρηστικότητα της προτεινόμενης μεθοδολογίας σε πρακτικά περιβάλλοντα. Από τις μελέτες προσομοίωσης έως τις εφαρμογές σε πραγματικά δεδομένα, η διατριβή επιδιώκει να γεφυρώσει τις έννοιες της στατιστικής και της μηχανικής μάθησης, με τη δεύτερη να αντλεί εργαλεία και μεθοδολογίες από την πρώτη, διαμορφώνοντας ένα ενοποιημένο πλαίσιο για τη στατιστική παρακολούθηση διεργασιών σε συνθήκες μεγάλων δεδομένων.

Table of contents

Nomenclature	xvii
List of figures	xxi
List of tables	xxvii
1 Introduction	1
1.1 A Literature Overview for Real-Time Monitoring	3
1.2 Bridging Statistical Process Monitoring and Machine Learning	5
1.3 Cluster Separability Indices for Real-Time Data Streams	6
1.4 Real-Time Monitoring using Maritime Data	7
1.5 Real-Time Monitoring using Weather Data	7
1.6 Evolution and Contribution of the Research Work	8
1.6.1 Publications Arising from this Thesis	10
1.7 Thesis Structure	10
2 A Literature Overview for Real-Time Monitoring	13
2.1 Introduction	13
2.2 Trends and Advancements	16
2.3 Statistical Approaches	20
2.3.1 Control Charts	20
2.3.2 Dimensionality Reduction and Regression Techniques	25
2.4 Machine Learning Approaches	28
2.4.1 Supervised Learning in Streaming Contexts	29
2.4.2 Unsupervised Learning and Anomaly Detection	31
2.4.3 Deep Learning for Temporal Sequence Modeling	32
2.4.4 Reinforcement Learning for Adaptive Monitoring	33
2.4.5 Online and Offline Learning	34
2.5 Conclusions	35

2.6	Summary	35
3	Bridging Statistical Process Monitoring and Machine Learning	37
3.1	Introduction	37
3.2	Motivation	39
3.3	Established Multivariate Monitoring Techniques	41
3.3.1	Control Charting	42
3.3.2	Estimating the In-Control Distribution	47
3.4	Clustering-Based Charts for Data Stream Monitoring	49
3.4.1	Creating Rational Subgroups in Data Streams	50
3.4.2	A Cluster Analysis Approach for Robust Data Stream Monitoring	51
3.5	Simulation and Performance Analysis	62
3.5.1	Simulation Setup	63
3.6	Results	75
3.6.1	Phase I: In-Control Conditions	77
3.6.2	Phase II: Mean Shift Detection	79
3.7	Conclusions	80
3.8	Summary	83
4	Monitoring Data Streams with Separability Indices	89
4.1	Introduction	89
4.2	Motivation	94
4.3	Materials and Methods	95
4.3.1	Core Principles and Implementation Phases	96
4.3.2	Clustering Data Streams in Real-Time	96
4.3.3	Clustering Validity Indices	97
4.3.4	Monitoring using Cluster Validity	102
4.4	Simulation: Cases Evaluated	104
4.4.1	Case 1: Drifting the Mean Vector (Shift in Location)	105
4.4.2	Case 2: Scaling the Data Points (Change in Spread)	108
4.4.3	Simulation Results	110
4.4.4	Simulation Conclusions	110
4.5	Real-World Data	113
4.5.1	The Palmer Penguins Dataset (Batch Data)	114
4.5.2	mHealth Dataset (Streaming Data)	118
4.6	Conclusions	124
4.7	Summary	125

5	Monitoring Applications in Maritime Data: Environmental Profiling and Engine Classification	127
5.1	Introduction	127
5.2	Motivation	129
5.3	Assessing Environmental Performance of Vessels	132
5.3.1	Theoretical Background	133
5.3.2	Materials and Methods	134
5.3.3	Analysis and Results	137
5.3.4	Conclusions	152
5.4	Ship Engine Model Selection	153
5.4.1	Materials and Methods	155
5.4.2	Data Collection	156
5.4.3	Application	162
5.4.4	Classification Results	174
5.4.5	Conclusions	181
5.5	Conclusion	183
5.6	Summary	196
6	Monitoring Applications in Hydrological Data: LSTM-Based Precipitation Forecasting	199
6.1	Introduction	199
6.2	Motivation	201
6.3	Materials and Methods	202
6.3.1	Data Preprocessing	203
6.3.2	Machine Learning Prediction Algorithm	206
6.3.3	Univariate Approach: Implementation Details	209
6.3.4	Multivariate Approach: Implementation Details	209
6.4	Results	209
6.4.1	Data Description	210
6.4.2	Data Preprocessing	211
6.4.3	LSTM Model Performance	214
6.4.4	Predicting	221
6.5	Conclusions	226
6.6	Summary	227
7	Closing Remarks	229
7.1	Discussion	229

7.2	Future Horizons	230
-----	---------------------------	-----

Nomenclature

Acronyms / Abbreviations

CO_2	Carbon Dioxide
ADWIN	ADaptive WINdowing
AIS	Automatic Identification System
ANN	Artificial Neural Network
CA	Cluster Analysis
CPS	Cyber-Physical Systems
CSI	Clean Shipping Index
CUSUM	Cumulative Sum
CVIs	Cluster Validity Indices
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DDM	Drift Detection Methods
DL	Deep Learning
DQN	Deep Q Networks
DRL	Deep Reinforcement Learning
DTs	Decision Trees
EC	European Commission
EEDI	Energy Efficiency Design Index

EEOI	Energy Efficiency Operational Indicator
ESI	European Commission
EVDI	Existing Vessel Design Index
EWMA	Exponentially Weighted Moving Average
GHG	Greenhouse Gas
HIPAA	Health Insurance Portability and Accountability Act
ICA	Independent Component Analysis
IFs	Isolation Forests
IMO	International Maritime Organization
IoT	Internet of Things
kNN	Decision Trees
KPCA	Kernel Principal Component Analysis
LCL	Lower Control Limit
LSTM	Long Short-Term Memory
LTER	Palmer Long-Term Ecological Research
MAE	Mean Absolute Error
MAR	Missing At Random
MCAR	Missing Completely At Random
MCUSUM	Multivariate Cumulative Sum
MEWMA	Multivariate Exponentially Weighted Moving Average
MIA	Multivariate Image Analysis
MICE	Multiple imputation by Chained Equation
ML	Machine Learning
MPCA	Moving Principal Component Analysis

MSE	Mean Squared Error
MSPCA	Multiscale Principal Component Analysis
MSPLS	Multiscale Partial Least Squares
MSPM	Multivariate Statistical Process Monitoring
NLP	Natural Language Processing
PCA	Principal Component Analysis
PCs	Principal Components
PHT	Page-Hinkley Test
PLS	Partial Least Squares
PPO	Proximal Policy Optimization
RL	Reinforcement Learning
RMSE	Root Mean Squared Error
RTM	Real-Time Monitoring
SEEMP	Ship Energy Efficiency Management Plan
SGD	Stochastic Gradient Descent
SIN	Ship Identification Number
Skape	Proposed Separability Index Family
SLSTM	Seasonal Long Short-Term Memory
SML	Statistical Machine Learning
SPM	Statistical Process Monitoring
SVMs	Support Vector Machines
tSNE	t-Distributed Stochastic Neighbor Embedding
UCL	Upper Control Limit
UMAP	Uniform Manifold Approximation and Projection

List of figures

2.1	The most popular subject areas for RTM publications, based on Scopus data. Note that some publications are classified under multiple subject areas, so the counts may not match the total number of publications per year.	17
2.2	Geographical perspective for RTM publications based on Scopus.	17
2.3	Publications for RTM based on Scopus.	18
2.4	An example of Control Chart ($\bar{X}$)	21
2.5	An example of Principal Component Analysis	26
2.6	An example of SVMs	30
2.7	An example of kNN (with $k = 5$ closest counterparts)	31
3.1	Scatter plot of Hotelling's T^2 statistic in two-dimensional space showing in-control (purple) and out-of-control (black) points. The dashed green ellipse depicts the UCL.	45
3.2	Control chart of Hotelling's T^2 statistic across time. Values surpassing the control limits signify process deviations requiring investigation. . .	45
3.3	Sequential samples extracted from a data stream following a MND. The blue points represent the S_n sample, while the orange points correspond to the subsequent S_{n+1} sample.	51
3.4	Visualization of Dunn Index parameters in two-dimensional space, showing inter-cluster and intra-cluster distances.	54
3.5	Dunn Index control chart over time, with the red dashed line indicating the UCL.	55
3.6	Illustration of the fundamental geometric quantities associated with the <i>Skape</i> family of indices. The solid black line indicates the distance between the two cluster centroids, while the dashed line represents the maximum pairwise distance between the two most distant points across the clusters.	59

3.7	Control chart based on the SK_{max}^* index over sequential periods. The red dashed line marks the UCL, which signifies possible process instability if exceeded, thereby prompting further investigation and the implementation of corrective measures.	60
3.8	Hadamard product of Multivariate Normal and Rayleigh Distributions under various degrees of correlation, illustrating the structural transformation as inter-variable correlation increases.	67
3.9	Hadamard product of Multivariate Normal and Poisson Distributions under various degrees of correlation, highlighting the discretization effects and structural deviations from standard multivariate normality.	69
3.10	Illustration of a mean shift from the initial sample (represented by dashed contour ellipse) to the shifted sample (represented by solid contour ellipse). The length of the shift vector (depicted as a magenta line) illustrates the magnitude of the shift $\mathbf{s}$ toward the 75 th percentile location in the feature space.	72
3.11	Inter-cluster pairwise distances in Multivariate Normal Distribution across d and ρ , illustrating contraction of means/minima and broadening of extremes with correlation.	78
3.12	Intra-cluster pairwise distances (Cluster 1) in Multivariate Normal Distribution across d and ρ , showing cohesion changes and heteroscedasticity with correlation.	79
3.13	Mean shift detection performance in Multivariate Normal Distribution across d and ρ , highlighting dimensional gains at low correlation and performance collapses in higher correlation.	80
3.14	Mean shift detection performance in Hadamard product of Rayleigh–Normal across d and ρ , showing rapid performance loss with correlation.	81
3.15	Mean shift detection performance in Hadamard product of Poisson–Normal across d and ρ , illustrating early performance collapse with growing correlation.	82
4.1	Visualization of a_i and b_i for a selected data point (represented by the red diamond marker). The blue dashed line (a_i) represents the mean intra-cluster distance (cohesion), while the green dashed line (b_i) shows the mean nearest-cluster distance (separation).	98

- 4.2 Visualization of between-cluster and within-cluster dispersion components in the Calinski-Harabasz Index framework. The purple star represents the overall data centroid. Blue dashed lines illustrate between-cluster spread (separation) measured from the overall centroid to each individual cluster centroid. Green dotted circles illustrate within-cluster spread (compactness) around each cluster centroid, with radii proportional to the internal point dispersion. 100
- 4.3 Visualization of cluster compactness and separation in the DBI framework. Green dashed circles around each cluster centroid illustrate within-cluster dispersion (s_i). Red solid lines connect each cluster to its most similar neighbor, with distances (d_{ij}) representing inter-cluster separation. The red line, labeled with largest R_{ij} , highlights the cluster pair with the absolute highest ratio of within-cluster to between-cluster distances, indicating the worst separation in the dataset. 101
- 4.4 Visualization example of 10 successive randomly sampled clusters with cumulative mean displacement. Each cluster contains 100 observations sampled from a bivariate normal distribution with unit covariance structure. The mean undergoes progressive displacement at each stage by adding an increment proportional to the stage index along a fixed direction vector. Convex hulls delineate the spatial boundary of each cluster, with color and transparency gradually transitioning to reflect temporal progression. The black dashed trajectory line and directional arrow trace the mean displacement path from initial to final stage. 106
- 4.5 Two uniformly distributed clusters with the reference cluster centered on zero mean vector and the displaced cluster (red) positioned at $x = y = 0.25$. Both clusters exhibit equal spread parameter of 1. 107
- 4.6 Progressive increase in data dispersion from initial sample (black dots) to modified sample (magenta dots), both generated from multivariate normal distribution. The impact of the scaling factor demonstrates the magnitude of dispersion increase after approximately 100 simulation iterations. 108
- 4.7 Progressive increase in data dispersion from initial sample (black dots) to modified sample (magenta dots), generated from multivariate uniform distribution. The impact of the scaling factor demonstrates the magnitude of dispersion increase after approximately 100 simulation iterations. 109

4.8	A photo taken by Dr. Kristen Gorman illustrating the three species of penguins in the dataset. Adélie (<i>Pygoscelis adeliae</i>), Chinstrap (<i>Pygoscelis antarcticus</i>), and Gentoo (<i>Pygoscelis papua</i>)	116
4.9	Intuitive look at how bill length, bill depth, flipper size, and body mass differ across species.	117
4.10	Correlation matrix of mHealth Dataset	121
5.1	A conceptual diagram of the monitoring procedure for assessing a ship's environmental compliance. Static and dynamic variables are fed into a ML framework to produce a final performance and compliance status. .	128
5.2	Framework of the analysis process.	135
5.3	Correlation matrix of dataset.	139
5.4	Mahalanobis Distance and Outlier Detection.	139
5.5	Scree plot for principal components.	140
5.6	Biplot showing variables' relation to the first two principal components (based on reduced dataset).	141
5.7	Base graphical tag representing environmental performance.	143
5.8	Representative graphical tags for ship environmental performance. . . .	144
5.9	Silhouette Score for cluster numbers analyzed by k-means on the three principal components.	145
5.10	Gap Statistic for cluster numbers analyzed by k-means on the three principal components.	146
5.11	Two clusters on the plane defined by first and second PCs.	146
5.12	Two clusters on the plane defined by first and third PCs.	147
5.13	Two clusters on the plane defined by second and third PCs.	147
5.14	Four clusters on the plane defined by first and second PCs.	148
5.15	Four clusters on the plane defined by first and third PCs.	148
5.16	Four clusters on the plane defined by second and third PCs.	149
5.17	Centroid values of the three principal components for the four k-means clusters.	150
5.18	The 4 main steps of the implementation process.	155
5.19	Spearman's correlation coefficients between the independent variables. .	167
5.20	Left plot: Scree plot of PCs for both the imputed and non-imputed datasets. Right plot: Variance explained by PCs for both the imputed and non-imputed datasets.	171

5.21	The 5 most frequent engine types in the reduced two-dimensional space obtained by the three dimensionality reduction methods (upper left plot: PCA, upper right plot: UMAP, bottom plot: t-SNE).	173
5.22	Performance of evaluation metrics (balanced accuracy) for imputed data using k-fold cross validation.	179
6.1	A conceptual diagram of the ML framework for predicting annual rainfall to support hydropower management in Greece.	201
6.2	The first feature vector $\mathbf{X}$ and target values $\mathbf{Y}$ of an SLSTM network with $s = 12$, $k = 2$, $m = 11$, and $f = 12$	208
6.3	The Peloponnese and the Ionian islands meteorological stations network (denoted with red dots) of the Institute for Environmental Research and Sustainable Development of the National Observatory of Athens. . . .	212
6.4	Percentage of missing values in the meteorological stations that were in operation for at least 4 years during the period under study and were active in 2020. The red dotted line facilitates the identification of the stations with more than 10% of missing values during their operation period.	212
6.5	Monthly precipitation time series plots for Megalopoli and Lykochia. In the embedded plot, the year 2018 is presented in detail.	214
6.6	The LSTM neural network architecture used in the analysis.	215
6.7	The boxplots of the predictions for each month for the Monemvasia (left plot) and Alagonia (right plot) stations based on the multivariate approach. The corresponding predictions based on the univariate approach are denoted with a blue circle. The true measured values for the two stations for each month are denoted with orange circles.	222
6.8	The kernel density estimates for the annual total precipitation (sum of 12 months' predictions) for all stations. The clusters of the stations according to their climate zone are denoted with brackets. The white lines in the density plots indicate the first and the third quartile. The blue dots represent the observed values of the total precipitation for each station in 2021.	225

List of tables

2.1	Contribution of Statistical and Streaming ML Methods for RTM	16
2.2	Contribution of Control Charts for RTM	21
2.3	Comparison of Dimensionality Reduction Methods for RTM	28
2.4	Contribution of ML Methods for RTM	35
3.1	Directional trends in pairwise distance statistics with increasing ρ , supporting the metric responses discussed in the text.	77
3.2	Multivariate Normal for different correlation cases across dimensions . .	86
3.3	Hadamard Product: Rayleigh - Normal for different correlation cases across	87
3.4	Hadamard Product: Poisson - Normal for different correlation cases across dimensions	88
4.1	AAC in Location Shift (Multivariate Normal Distribution)	111
4.2	AAC in Spread Change (Multivariate Normal Distribution)	112
4.3	AAC in Location Shift (Multivariate Uniform Distribution)	112
4.4	AAC in Spread Change (Multivariate Uniform Distribution)	113
4.5	Penguin Classification Result (where, Class A is Adelie, C is Chinstrap, and G is Gentoo).	118
4.6	Activity set with durations and repetition counts	120
4.7	Metric Performance for mHealth Dataset (subject1)	123
5.1	Data Set Description.	137
5.2	PCA loadings for the first three PCs on initial and reduced datasets with pairwise differences.	141
5.3	Pearson correlation coefficients with p-values (in parentheses) among PCs within each identified cluster.	151
5.4	PCA loadings for the first three PCs, including pairwise differences. . .	152
5.5	Description of each subset of the data set, based on data providers. . .	164

5.6	First group of variables (exterior measurements of a ship).	165
5.7	Second group of variables (operational).	165
5.8	Evaluating metrics of MICE imputation, considering the Bayesian Ridge estimator as a parameter for the imputation algorithm.	169
5.9	Evaluating metrics of MICE imputation, considering the Light Gradient Boosting estimator as a parameter for the imputation algorithm.	169
5.10	Evaluating metrics of MICE imputation, considering the Extreme Gradient Boosting estimator as a parameter for the imputation algorithm.	170
5.11	Loadings of PCA for the imputed dataset.	172
5.12	The final main engine models of the procedure along with their encoding and frequency in the dataset.	174
5.13	Performance of evaluation metrics (balanced accuracy) for imputation data and the top performance for each of the three dimensionality reduction methods using k-fold cross validation. The time taken for each algorithm is also included.	177
5.14	The average balanced accuracy scores from five iterations of the k-fold cross-validation technique of the ExtraTreesClassifier across different n_estimators values (for imputed and non-imputed data as well as for multiple dimensionality reduction methods).	180
5.15	Confusion matrix of the 5 most frequent engine models (imputed data) and n_estimator = 50.	181
5.16	<i>Cont.</i> Performance of evaluation metrics (balanced accuracy) and time taken for each algorithm.	184
5.17	Performance of evaluation metric (balanced accuracy) for imputation data using PCA method for different components using “train_test_split”, along with the time taken for training each algorithm.	185
5.18	Performance of evaluation metrics (balanced accuracy) for imputation data using t-SNE with 2 components and different perplexity values using “train_test_split”, along with the time taken for training each algorithm.	186
5.19	Performance of evaluation metrics (balanced accuracy) for imputation data using t-SNE with 3 components and different perplexity values using “train_test_split”, along with the training time for each algorithm.	187

5.20	Performance of evaluation metric (balanced accuracy) for imputation data using UMAP with 2 components and different near-neighbor values using “train_test_split”, along with the time taken for training each algorithm.	188
5.21	Performance of evaluation metric (balanced accuracy) for imputation data using UMAP with 3 components and different near-neighbor values using “train_test_split”, along with the time taken for training each algorithm.	189
5.22	Performance of evaluation metric (balanced accuracy) for imputation data using the PCA method for different components using k-fold cross validation, along with time taken for training each algorithm.	190
5.23	Performance of evaluation metrics (balanced accuracy) for imputation data using t-SNE with 2 components and different perplexity values using k-fold cross validation, along with the time taken for training each algorithm.	191
5.24	Performance of evaluation metric (balanced accuracy) for imputation data using t-SNE with 3 components and different perplexity values using k-fold cross validation, along with the time taken for training each algorithm.	192
5.25	Performance of evaluation metrics (balanced accuracy) for imputation data using UMAP with 2 components and different near-neighbor values using k-fold cross validation, along with the time taken for training each algorithm.	193
5.26	Performance of evaluation metrics (balanced accuracy) for imputation data using UMAP with 3 components and different near-neighbor values using k-fold cross validation, along with the time taken for training each algorithm.	194
6.1	General representation of feature vector $\mathbf{X}$ and target values $\mathbf{Y}$ for an SLSTM network. Here, n is the total number of observations, s is the seasonal period, k the number of seasonal lags, m the max lagged values, and f the forecast horizon.	208
6.2	Grid search for the univariate approach using the Amaliada station. The smallest MSE value is marked in bold.	218
6.3	The CPU usage range and the time required to run the analysis for the selected hyperparameters.	219

- 6.4 Grid search for the multivariate approach using the initial split. The smallest MSE value is marked in bold. 220
- 6.5 The monthly actual precipitation values (in mm) and the predictions using the recently proposed algorithm and the prediction for 12 months ahead based on the best SARIMA model. In the parentheses, the absolute differences between each prediction from the actual value are reported. The minimum for each month and each station are marked in bold. 224

Chapter 1

Introduction

In the past two decades, the proliferation of digital technologies has fundamentally redefined how data are generated, stored, and utilized across virtually all sectors of human activity. From intelligent industrial systems and autonomous manufacturing platforms to social media ecosystems and healthcare infrastructures, data streams of unprecedented scale, granularity, and complexity are continuously produced. This development, now widely recognized as the era of *big data*, has had a transformative impact not only on the scientific and engineering domains but also on the broader socioeconomic fabric. The defining characteristics of big data are conventionally summarized by the “five Vs”: volume, velocity, variety, veracity, and value [1, 2]. Together, these dimensions encapsulate the inherent analytical challenges in processing modern data ecosystems.

Beyond these canonical dimensions, additional “Vs” such as *variability* and *visualization* emphasise temporal irregularity and the need to render complex data intelligible for human operators. While classical statistical methods were developed primarily for bounded, well curated datasets, big data applications are frequently characterized by distribution across heterogeneous infrastructures, real time processing demands, and dynamic evolution [3]. Addressing these challenges requires not only algorithmic innovation but also new paradigms in data storage, computational frameworks, governance, and human–machine interaction.

The emergence of big data has coincided with Industry 4.0, marked by the fusion of physical and digital infrastructures through cyber–physical systems. Sensors, actuators, and smart devices embedded within industrial machinery, transportation networks, and energy facilities collectively form the backbone of the Internet of Things (IoT), generating continuous data streams that reflect the operational and environmental state of complex systems [3]. At the same time, social and consumer centric systems generate

digital footprints through social media, e-commerce, mobile devices, and wearables. These heterogeneous, high velocity data sources create an integrated yet complex ecosystem that demands analytical methods capable of operating across distributed infrastructures while meeting strict constraints on latency, reliability, and resource efficiency.

Within this environment of continuous data flows, anomaly detection has emerged as an issue of critical importance. An anomaly is generally understood as any observation, subsequence, or structural pattern that deviates substantially from what is considered normal behaviour. In practice, anomalies often signal significant occurrences with economic, operational, or societal implications, ranging from fraud in financial transactions and incipient equipment failures in manufacturing to cyber attacks in networked systems and early signs of extreme meteorological events [4, 5, 6]. Anomaly detection can therefore be regarded as a computational analogue of human attentional mechanisms, directing focus towards unexpected signals and enabling timely intervention.

Modern applications, however, rarely involve static, low dimensional datasets. Instead, they operate in streaming environments where data arrive as unbounded, temporally ordered sequences that cannot be stored in full. These streams are subject to concept drift, with underlying data generating mechanisms evolving over time, and they often exhibit high dimensionality, where hundreds or thousands of variables are observed simultaneously. In such settings, many traditional anomaly detection techniques become impractical or unreliable due to storage limitations, latency constraints, instability of high dimensional distance measures, and violations of distributional assumptions [7]. This motivates the development of real time monitoring frameworks that are both statistically principled and computationally scalable.

The present thesis seeks to advance the state of anomaly detection methodologies for high dimensional streaming data by developing clustering based real time monitoring frameworks that utilize Clustering Validity Indices (CVIs) as adaptive indicators. These frameworks address inherent limitations of classical Multivariate Statistical Process Monitoring (MSPM) in nonstationary and complex environments. The contributions are theoretical, methodological, and empirical, demonstrating improvements in detection accuracy, scalability, and interpretability in maritime and environmental case studies.

1.1 A Literature Overview for Real-Time Monitoring

Real-time monitoring (RTM) has matured into a multidisciplinary field that blends classical statistical thinking with modern data driven modelling in order to support timely and reliable decisions. At its core, RTM aims to detect departures from expected behaviour under constraints that are uncommon in traditional offline analysis, such as continuous arrival of observations, limited storage, latency requirements, and frequent violations of stationarity. These constraints shape both the modelling choices and the evaluation criteria, because a method must be not only accurate but also computationally stable, interpretable to practitioners, and robust to evolving operational conditions.

Historically, RTM has its roots in early twentieth century quality control, with Shewhart charts providing a prototypical framework for univariate process monitoring. Subsequent developments introduced Exponentially Weighted Moving Average (EWMA) and Cumulative Sum (CUSUM) designs to improve sensitivity to small and sustained shifts. As industrial processes became more complex and multivariate, MSPM extended these ideas to multiple correlated variables, notably through Hotelling's T^2 statistic and related control charts. In parallel, data mining and Machine Learning (ML) communities contributed distance based, density based, and clustering based anomaly detection methods that relax strict parametric assumptions and rely instead on geometric or structural properties of the data [8, 9, 10].

In recent decades, ML techniques have broadened the methodological landscape. Supervised, semi supervised, and unsupervised models, including Support Vector Machines (SVMs), Tree Based Ensembles, Autoencoders, Isolation Forests (IFs), and other learners, have been adapted to anomaly detection with varying degrees of reliance on labelled data [11, 12, 13]. Deep learning (DL) architectures such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTMs), Variational Autoencoders (VAEs), and Generative Adversarial Networks (GANs) further extend the capability to model high dimensional, sequential, and multimodal data. At the same time, these advances have raised concerns about interpretability, robustness under concept drift, computational costs, and fairness in critical applications [14, 15]. Hybrid frameworks that integrate statistical rigour with ML flexibility have therefore attracted growing interest, aiming to preserve interpretability while exploiting data driven adaptivity.

Against this background, a modern overview of RTM must connect foundational MSPM concepts with contemporary ML and DL approaches, highlighting their respective strengths and limitations and clarifying where new developments are needed. This broader perspective motivates the methodological developments and application studies that follow, each contributing a piece of the overall RTM pipeline from representation to detection and decision support.

Many contemporary monitoring problems are inherently high dimensional, with strong dependence structures and heterogeneous sources that challenge classical multivariate techniques. In such settings, standard covariance estimation may become unstable, distance measures can concentrate, and traditional control limits may lose reliability, particularly when the process distribution changes over time. Anomalies may manifest only through subtle inter variable interactions that are invisible at the level of marginal distributions, and many algorithms scale poorly with the dimension of the feature space [7].

Effective monitoring in high dimensions therefore requires careful representation choices that preserve information relevant to change detection while reducing redundancy and noise. Dimensionality reduction techniques, spanning linear methods such as Principal Component Analysis (PCA) and Independent Component Analysis (ICA) to nonlinear embeddings such as t-Distributed Stochastic Neighbor Embedding (t-SNE) and Uniform Manifold Approximation and Projection (UMAP), have been used to construct lower dimensional manifolds that retain essential structure. Feature selection strategies and neural network based representation learning, including autoencoders, provide complementary approaches for identifying compact, task relevant summaries of the data.

At the same time, streaming operation imposes additional constraints. Algorithms cannot rely on full historical storage and must operate under strict memory and latency budgets, processing data incrementally as it arrive. Concept drift requires models that adapt continuously to evolving patterns of normal behaviour, while maintaining controlled false alarm rates and interpretability. The methodological focus in this thesis is on constructing monitoring statistics that remain sensitive to meaningful changes in such environments, even as dimensionality, correlation, and non Gaussian structure complicate the geometry of the data.

1.2 Bridging Statistical Process Monitoring and Machine Learning

A central theme in modern RTM is the integration of Statistical Process Monitoring (SPM) with ML methodologies. SPM provides a rigorous, interpretable foundation for detecting departures from normal behaviour, with well-established principles for reference-based operation, error control, and the construction of actionable monitoring statistics. These strengths are particularly valuable in regulated and safety-critical domains, where transparency and accountability are essential. However, SPM traditionally relies on strong assumptions about data distributions and stationarity, which can be violated in complex, high-dimensional, and rapidly evolving environments.

On the other hand, ML offers flexible, data-driven mechanisms for capturing non-linear relationships, heterogeneous feature types, and dynamic system behaviour. ML techniques can adapt to evolving data generating mechanisms and are less constrained by parametric assumptions, making them suitable for complex, real-world settings. Yet, ML approaches often sacrifice interpretability and formal guarantees, and can behave as “black boxes” that are difficult to justify in practice.

Bridging SPM and ML involves creating hybrid frameworks that leverage the strengths of both paradigms. This typically requires the design of intermediate representations and summary statistics that connect learned data structure to monitoring logic. One promising approach is to use unsupervised learning, particularly clustering, as a structural layer between raw data and monitoring logic. In this view, typical operating regimes are captured through clusters or latent representations, and departures from normality are assessed via changes in cluster geometry, separability, or temporal stability, rather than through fixed distributional models.

The resulting summary measures—such as cluster validity indices or separability statistics—can serve as interpretable monitoring statistics that are compatible with SPM-style reasoning, while still reflecting complex dependence patterns learned from the data. This integrated perspective enables the design of RTM pipelines where modelling, representation, and decision rules are developed jointly, allowing statistical tools to guide threshold selection and performance evaluation, while ML components supply flexible, adaptive descriptions of system behaviour. The framework aims to balance sensitivity to subtle changes, robustness under concept drift, computational feasibility in streaming environments, and the level of interpretability required for practical deployment.

1.3 Cluster Separability Indices for Real-Time Data Streams

Clustering provides a natural and interpretable approach to monitoring when labels are scarce and when normal behaviour is better described through data structure than through strict parametric assumptions. In this perspective, the monitoring problem is reframed as the stability of cluster structure over time, where changes in separation, compactness, or relative geometry signal potential distributional shifts. Cluster separability and validity indices—such as the Silhouette, Dunn, Calinski–Harabasz, Davies–Bouldin, and the Skape family—can be repurposed as RTM statistics, computed repeatedly on sequential data blocks.

These indices quantify different aspects of cluster quality, including intra-cluster cohesion and inter-cluster separation, and their continuous evaluation can enhance anomaly detection by monitoring evolving cluster structures. For example, shifts in location, changes in spread, or alterations in dependence patterns can be detected by tracking index values over time. The practical challenge is to ensure stability under short windows and streaming noise, so that the resulting signals are not dominated by sampling variability. This motivates careful choices regarding window size, rational subgrouping, updating strategy, and decision thresholds to obtain monitoring schemes that remain reliable in continuous operation.

Moreover, the interpretability of these indices is a significant advantage: each index provides a clear geometric or statistical interpretation, allowing practitioners to understand why a signal has triggered and how it relates to the underlying data structure. This is particularly valuable in operational settings where monitoring outputs must be actionable and justifiable. The thesis develops clustering-based monitoring frameworks that use these indices as adaptive indicators, providing robust and interpretable solutions for high-dimensional, streaming data environments.

Cluster separability indices also facilitate the comparison of different clustering results and can guide the selection of optimal cluster numbers. In RTM, this enables the detection of structural anomalies that manifest through alterations in clustering patterns, supporting interpretability by associating anomalies with specific cluster memberships or transitions. The thesis further explores the performance of these indices under various conditions, including high dimensionality, strong correlation, and non-Gaussian distributions, highlighting their robustness and limitations in practical applications.

1.4 Real-Time Monitoring using Maritime Data

Maritime monitoring exemplifies the realities of operational RTM, characterized by heterogeneous measurements, missing data, dependence among variables, and stringent domain constraints tied to safety, efficiency, and regulatory compliance. In such an environment, monitoring objectives frequently entail a combination of descriptive tasks, such as the profiling of operational behavior and environmental performance, with predictive tasks, such as the inference of unobserved attributes from partial records.

Consequently, effective solutions must integrate preprocessing, robust representation learning, and statistical decision rules in a manner that facilitates timely interventions. Multivariate reduction and clustering can be used to construct interpretable profiles of vessel and voyage characteristics that support environmental assessment and compliance evaluation. Concurrently, the employment of supervised learning techniques has the capacity to infer engine-related attributes or other technical specifications from incomplete datasets. This, in turn, enhances the quality and completeness of monitoring information available to stakeholders.

From a methodological perspective, maritime data underscore the imperative of managing multivariate structures while preserving interpretability for operators, auditors, and regulators. This is due to the fact that monitoring outputs are frequently employed for planning and compliance-related initiatives. This application setting underscores the necessity for validation across subsets of vessels and operational regimes, as models trained on one segment may not directly transfer to another. Consequently, maritime monitoring serves as an exacting testbed for end-to-end RTM pipelines, necessitating both precision and operational credibility.

1.5 Real-Time Monitoring using Weather Data

The practice of weather and precipitation monitoring introduces a complementary RTM challenge, namely, temporal dependence and forecasting under uncertainty. In this context, decisions are influenced by both short-term dynamics and longer-term variability. In this setting, time series may manifest seasonality, regime changes, and spatial dependence across networks of stations. However, such series are also susceptible to missing values and measurement noise. An effective modeling approach must capture temporal structure without sacrificing responsiveness, and it must provide stable predictions that can be aggregated for downstream planning tasks.

Sequence learning models based on recurrent neural network architectures, including LSTM networks, offer a natural framework for modeling such dependencies. By training on historical precipitation records from multiple stations, these models can exploit both temporal and spatial coherence to improve forecast skill. Concurrently, operational utilization necessitates meticulous evaluation due to the necessity of interpreting predictive performance within the framework of its implications for hydropower management and associated applications.

This setting underscores the role of RTM as a decision support tool, where the modeling of choices and uncertainty summaries are pertinent to the processes of planning and resource allocation. It complements the maritime case study by highlighting different aspects of the RTM problem, in particular the interaction between temporal forecasting, spatial structure, and real-time operational constraints in environmental systems.

1.6 Evolution and Contribution of the Research Work

The initial idea behind this research emerged from the need to build a framework for monitoring multivariate data streams. Existing SPM methods rely heavily on MSPM techniques, where the Hotelling's T^2 statistic serves as a fundamental metric for detecting changes. However, the T^2 statistic assumes normality of the underlying data distribution. Moreover, the estimation of its control limits through simulation requires parameter estimation—most notably, the computation of covariance matrices—which becomes computationally expensive and statistically unreliable in non-normal or high-dimensional data environments. In the era of big data, such parametric approaches become impractical, creating the need for non-parametric alternatives.

The first step of this dissertation was therefore to explore simulation-based procedures for determining control limits for non-parametric metrics. The second step focused on investigating existing non-parametric measures that could yield reasonable performance in capturing distributional changes. The third step involved performance evaluation and comparative analysis to derive valid conclusions on the efficiency and applicability of these methods.

Having defined these directions, the core objective of this dissertation became the design and development of a novel non-parametric metric for statistical monitoring of data streams. An appropriate and promising research field for such development lies within statistical ML, more specifically in the area of unsupervised learning —

clustering. Metrics used to assess the quality and separability of clusters, known as CVIs, provided a natural foundation for the proposed approach. Among the established CVIs, the Dunn Index was selected as a baseline due to its conceptual clarity and interpretability.

The experimental process began by constructing synthetic data streams generated from normal distributions, for which the T^2 statistic and its parameters were first computed, followed by the determination of control limits (Phase I). Subsequently, a shift was introduced at the 75th percentile to simulate a process change (Phase II). However, the preliminary results revealed that the Dunn Index struggled to detect changes effectively when clusters exhibited high overlap. This limitation motivated the development of a novel metric capable of overcoming the weaknesses of the Dunn Index.

Building upon the distance-based foundations of Dunn’s formulation and after extensive experimentation and refinement, a new metric named **Skape** was introduced. Although initial comparisons with the T^2 statistic did not immediately demonstrate superiority, further testing on high-dimensional data streams revealed the strong potential of Skape in accurately detecting changes. Subsequent experiments extended to non-normal distributions confirmed its robustness and adaptability beyond Gaussian assumptions.

To ensure a comprehensive evaluation, the framework and the newly developed metric were further tested using real-world datasets, selected for their reliability, availability of ground truth labels, and frequent use in both academic and industrial benchmarking. Once adapted to these data streams (particularly in classification-oriented contexts), Skape exhibited impressive performance, often surpassing traditional approaches.

Finally, to complete the validation process, Skape was compared not only with the T^2 and Dunn metrics but also with a range of contemporary indices commonly employed in both academia and industry. Across these comparisons, Skape consistently demonstrated significant improvements in detection capability, computational efficiency, and robustness across varying data characteristics. Following these results, the final stage of this research involved documenting and formalizing these findings in the present dissertation, consolidating both the methodological innovation and its demonstrated contributions to the field of statistical process monitoring and ML.

1.6.1 Publications Arising from this Thesis

The work presented in this dissertation has led to a number of scientific publications, reflecting both its methodological contributions and its applied impact. Chapter 5 draws from two published peer-reviewed articles: “Assessing Ships’ Environmental Performance Using Machine Learning” in *Energies* **16**(6), 2544 [16] and “Ship Engine Model Selection by Applying Machine Learning Classification Techniques Using Imputation and Dimensionality Reduction” in *Journal of Marine Science and Engineering* **12**(1), 97 [17]. Chapter 6 is based on the article “Projecting Annual Rainfall Timeseries Using Machine Learning Techniques” published in *Energies* **16**(3), 1459 [18]. In addition to these published works, the core methodological contributions and simulation studies reported in Chapters ?? have been submitted as journal articles and are currently under review, covering the literature synthesis on RTM, the integration of MSPM with clustering-based methods, and the development and evaluation of separability-index-based monitoring schemes for high-dimensional data streams.

1.7 Thesis Structure

The subsequent chapters unfold as follows:

Chapter 2: Literature Overview This chapter examines the evolution of statistical techniques adapted for streaming and large-scale data, bridging advancements in ML to address uncertainty, high dimensionality, and dynamic environments. It provides a critical survey of current anomaly and change detection methods, highlighting methodological gaps and identifying promising areas for innovation.

Chapter 3: Monitoring High-Dimensional Data Streams Chapter 3 investigates the impacts of dimensionality and sparsity on process monitoring, illustrating the challenges inherent in interpreting high-velocity data. It proposes the integration of clustering methods with traditional statistical monitoring, demonstrating performance improvements over canonical approaches.

Chapter 4: Cluster Separability Indices for Real-Time Data Streams This chapter advocates for continuous quality oversight through CVIs such as the Silhouette, Dunn, Calinski-Harabasz, and Davies–Bouldin scores. Real-time evaluation of clustering results enhances anomaly detection sensitivity and computational resource efficiency, as demonstrated using synthetic case studies.

Chapter 5: Real-Time Monitoring using Maritime Data In addressing maritime data challenges, this chapter details preprocessing workflows, dimensionality reduction techniques, and classification methods applied to ship engine and environmental datasets, enabling robust assessment of environmental performance.

Chapter 6: Real-Time Monitoring using Weather Data The final chapter presents ML-based precipitation forecasting, which is critical for hydropower management. It examines model performance across various Greek climatic zones and discusses prediction biases observed in the empirical results.

Chapter 7: Closing Remarks The final chapter presents ML-based precipitation forecasting, which is critical for hydropower management. It examines model performance across various Greek climatic zones and discusses prediction biases observed in the empirical results.

Chapter 2

A Literature Overview for Real-Time Monitoring

2.1 Introduction

RTM systems are vital for identifying and reacting to changes, anomalies, and critical incidents across many domains such as industrial automation, finance, healthcare, cybersecurity, and environmental sensing. A key component within many RTM implementations is MSPM, which facilitates analysis across multiple correlated data streams to detect subtle shifts in system behavior [19]. RTM demands the real-time processing of fast, voluminous, and heterogeneous data streams, enabling quick feedback and adjustment to changing operating conditions.

This chapter provides a thorough review of both traditional statistical methods and modern ML techniques applied to RTM, with a particular focus on integrating MSPM methods with streaming data analytics. The methods are organized into statistical frameworks, including multivariate process monitoring and control charts, as well as ML approaches such as supervised, unsupervised, deep, and Reinforcement Learning (RL) models. Each category offers specific solutions addressing unique challenges inherent in RTM applications.

A bibliometric review of recent research outputs highlights that Computer Science, Decision Sciences, Engineering, Mathematics, Physics, and Astronomy are key contributors to RTM literature. This reflects the growing importance of intelligent, data-driven monitoring solutions in scientific and technological contexts. System considerations, including edge computing, distributed architectures, and advanced frameworks for data streams, play an increasingly crucial role in practical and scalable RTM deployments.

RTM involves the continuous observance and evaluation of system states or data streams as events happen, with the goal of detecting significant trends, fluctuations, or anomalies almost immediately. In contrast to batch processing, which analyzes historical data with substantial delays, RTM is designed to operate with minimal latency, frequently within milliseconds to seconds following data generation [20]. This rapid response is essential in domains where safety, reliability, or quick decision-making are critical.

RTM systems handle continuous, high-speed data flows from varied sources including sensors, financial markets, and networks. These data streams pose challenges in velocity (frequency of data), volume (scale of data), and variety (heterogeneous structures), requiring scalable, incremental, and memory-efficient algorithms [21, 22]. The necessity for real-time decisions often excludes deferred, offline analysis, necessitating immediate algorithmic processing.

Meeting these stringent needs demands specialized frameworks incorporating online learning, adaptive summarization methods like sketches and sliding windows, and parallel or distributed processing techniques. Edge analytics, which process data near the source, further enhance scalability and responsiveness, especially under strict latency limitations. The indispensable role of RTM is apparent in several sectors:

- **Cybersecurity:** RTM enables the early detection and mitigation of network intrusions, helping maintain system integrity and prevent breaches [23].
- **Healthcare:** Continuous monitoring of vital signs supports urgent clinical decisions, sometimes saving lives [24].
- **Manufacturing:** Real-time identification of equipment malfunctions helps minimize downtime and improve operational productivity [25].
- **Finance:** RTM assists in promptly spotting fraud, supporting asset protection and limiting financial loss [26].

Thus, RTM underpins timely, data-informed interventions that enhance both safety and operational performance in critical applications.

Historically, monitoring approaches have relied on fixed rules, static thresholds, and offline analyses, sufficient for systems with limited and simple data sources. However, the expansion of sensor networks, telemetry, and rapid financial data streams has exposed significant scalability and adaptability challenges in these methods [27]. A principal challenge in RTM is the precise, prompt recognition of anomalies and changes in dynamic, high-throughput systems. Progress addressing this includes:

- **SMP:** Methods like Shewhart, CUSUM, and EWMA charts.
- **Change Point Detection:** Using sequential and Bayesian algorithms.
- **Time Series Analysis:** Employing models such as ARIMA, state-space frameworks, and Kalman filters.
- **Online ML Algorithms:** Incremental approaches including Hoeffding Trees.
- **Concept Drift Detection:** Algorithms like Drift Detection Methods (DDM) and ADaptive WINdowing (ADWIN).
- **Signal Processing:** Utilizing wavelet transforms and sophisticated filters.

Choosing appropriate methods depends on data types, required response times, and computational constraints. Hybrid solutions balancing sensitivity with scalability are favored in practical streaming contexts.

Modern ML and statistical frameworks, such as MSPM, provide strong RTM solutions. MSPM enables the concurrent monitoring of multiple variables, fostering comprehensive understanding and adaptable responses to evolving states [19]. By learning directly from data, MSPM models exceed the limitations of rigid rule-based approaches, improving sensitivity and responsiveness [28].

Advancements in computation and streaming infrastructure have further empowered RTM systems. ML-enhanced frameworks, including MSPM, efficiently handle fast data streams, adapt to concept drift, and deliver low-latency predictions, notably at network edges [20, 21, 29]. Nevertheless, classical statistical models remain preferred where interpretability, formal assurance, and reliability are imperative [30].

Recent studies underscore the value of integrating statistical and ML techniques for RTM. While traditional approaches are prized for clarity and efficiency, they encounter difficulties with high-dimensional and evolving data. Methodologies feature dimensionality reduction techniques such as PCA and Partial Least Squares (PLS) alongside hybrid statistical-ML frameworks to boost performance in complex RTM environments.

Hybrid frameworks that integrate statistical principles with ML adaptability offer potential for reconciling these competing requirements in operational RTM systems, as outlined in Table 2.1.

The chapter is arranged as follows:

- Section 2.2 details the foundations and historical evolution of RTM.

Table 2.1 Contribution of Statistical and Streaming ML Methods for RTM

Aspect	Statistical	ML
Interpretability	High [19]	Medium–Low (model dependent) [28]
Scalability	Medium (batch/window-based limits)	High for streams [20]
Non-stationarity	Limited in classical methods; adaptive variants exist [19]	Strong (drift-aware, online adaptation) [28]
Computational Cost	Low	Moderate–High
Real-time Adaptability	Limited (requires re-estimation)	Excellent (instance-wise updates)
Data Variety	Low (typically structured, homogeneous data)	High (handles structured, unstructured, multi-modal data) [27]
Distribution Dependence	High (assumes specific distributions, e.g., Gaussian) [19]	Low (distribution-free, non-parametric methods) [28]

- Section 2.3 examines traditional statistical methods, their advantages, drawbacks, and recent improvements.
- Section 2.4 explores ML approaches for adaptive RTM in dynamic settings.
- Section 2.5 presents the conclusions, summarizing key findings.
- Section 2.6 provides a summary and outlines future directions for RTM research.

2.2 Trends and Advancements

RTM is described as the continuous process involving the acquisition, processing, and analysis of data with minimal latency, which permits prompt identification and response to dynamic events. This ability is fundamental across a wide range of academic disciplines, as demonstrated in Figure 2.1, and is reflected in significant research activity from many regions worldwide (refer to Figure 2.2). The increasing academic focus is also highlighted by the publication trends shown in Figure 2.3, which exhibits Scopus publication counts for the term “real-time monitoring” over the past ten years.

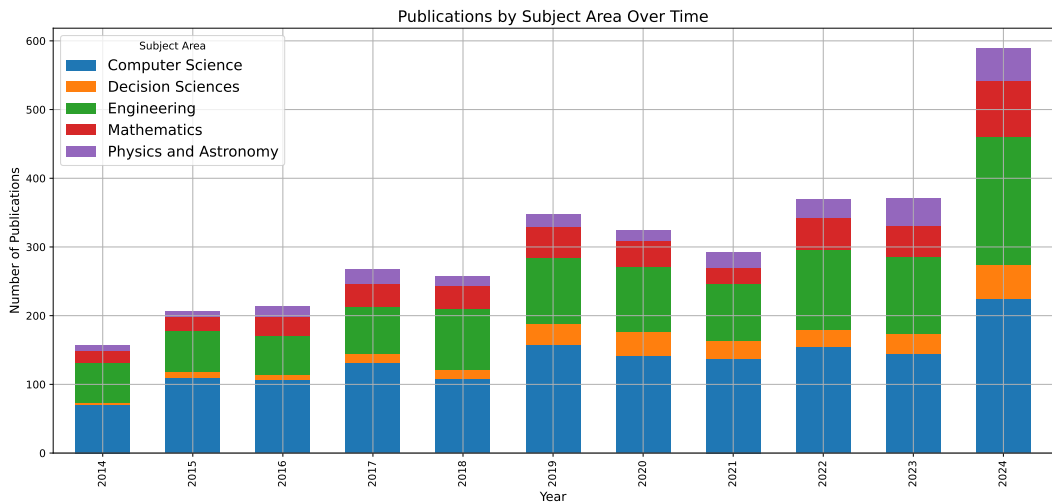


Fig. 2.1 The most popular subject areas for RTM publications, based on Scopus data. Note that some publications are classified under multiple subject areas, so the counts may not match the total number of publications per year.

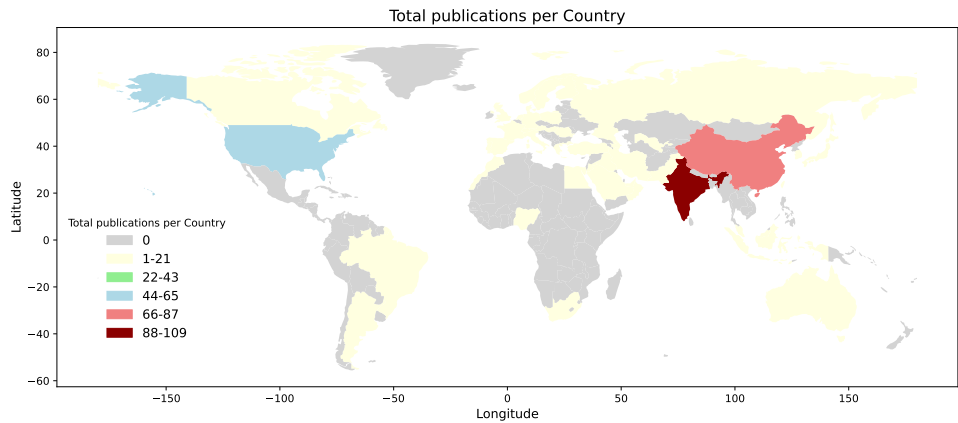


Fig. 2.2 Geographical perspective for RTM publications based on Scopus.

The origins of RTM date back to the early twentieth century with Walter A. She-
whart’s pioneering work on control charts in 1924, primarily aimed at monitoring
manufacturing processes. The concept progressed through the mid twentieth century,
gaining noteworthy traction within military and aerospace sectors, where collecting
and reacting to real-time data became critically important. Supported by funding

from the U.S. Navy, the Whirlwind system introduced dynamic radar tracking for air defense, representing a landmark development in real-time system evolution [31]. Over time, RTM matured within industrial process control and computer engineering. The 1970s and 1980s marked the broad implementation of programmable logic controllers (PLCs), which facilitated automated RTM in manufacturing environments. Concurrently, advances in real-time operating systems and embedded control logic supported telecommunications, avionics, and robotics fields [32].

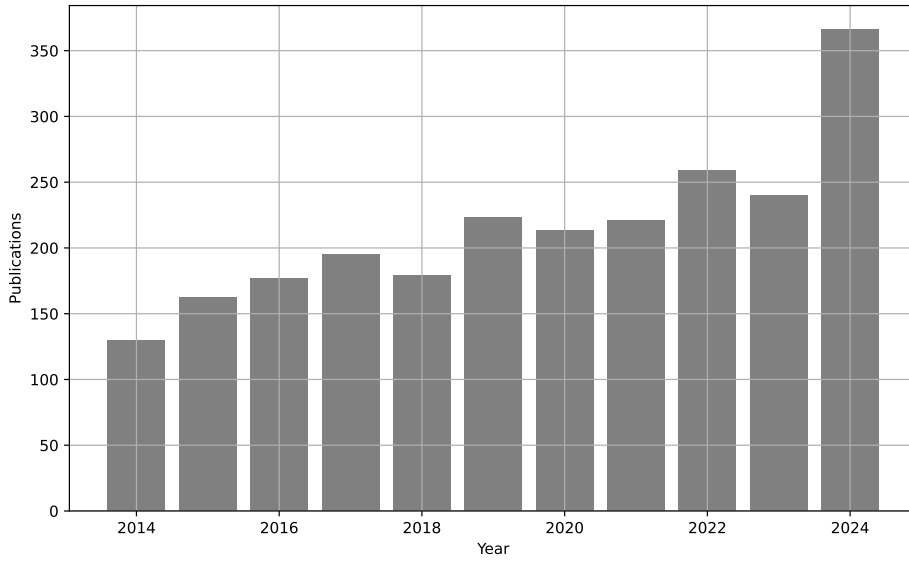


Fig. 2.3 Publications for RTM based on Scopus.

Within Computer Science, RTM encompasses the formulation of algorithms and design of system architectures optimized for data streaming and computations with low latency. Decision sciences leverage real-time data to facilitate swift, well-informed decisions under uncertainty. Engineering fields utilize RTM as a fundamental component for automation and control systems that require immediate feedback. Mathematics contributes essential theoretical frameworks and statistical methodologies supporting real-time inference and anomaly detection. Furthermore, RTM plays an important role in Physics and Astronomy, enabling the observation and rapid analysis of ephemeral phenomena such as cosmic events and particle interactions. While present-day RTM is commonly associated with AI-driven frameworks and the IoT, its foundational concepts and technologies have roots extending several decades, grounded in early innovations in computing and control theories.

The evolution of RTM within Decision Sciences arose from the necessity for timely and effective decisions in volatile and uncertain contexts. Initial uses surfaced in

operations research and management sciences in the 1960s and 1970s, where real-time data acquisition and examination started to aid aspects like inventory control, supply chain management, and emergency planning [33]. As streaming data grew prevalent towards the close of the twentieth century, techniques such as dynamic programming, stochastic optimization, and real-time simulations were refined to support adaptive decision frameworks capable of instantly responding to shifting system conditions [34]. The assimilation of RTM into decision support systems became vital in areas including finance, healthcare, and transportation, enabling continuous feedback loops that improve resource utilization and risk mitigation [35].

The mathematical underpinnings of RTM rest on developing dynamic models, statistical inference methods, and sequential analysis strategies that facilitate ongoing updates and decision processes. Since the 1950s, mathematicians have studied real-time signal processing and stochastic processes to model systems evolving amid noise and uncertainty [36]. The field of sequential analysis, established in the 1940s and 1950s, formalized frameworks for real-time hypothesis testing and change-point detection, remaining cornerstone techniques in anomaly detection and quality control today [37]. More recently, progress in online learning algorithms and adaptive filtering has expanded this mathematical toolkit, enabling incremental model updates within streaming data environments and ensuring robustness against concept drift and non-stationarity [38].

By the 1990s, RTM extended beyond engineering into scientific disciplines. For example, in astronomy, the advent of real-time sky surveys and gamma-ray burst detection systems empowered astronomers to react instantly to transient cosmic events [39]. Similarly, environmental monitoring advanced through sensor networks and remote sensing modalities to provide near real-time information on weather, seismic activity, and ecosystem patterns. These systems strengthen early warning capabilities and foster adaptable resource management [40]. This progression mirrors a broader trend to blend technological innovation and ecological insights to confront complex environmental challenges.

Research in RTM spans a broad spectrum of subjects, demonstrating the pronounced interdisciplinary nature of the field. The increasing demand for real-time data processing, analytics, and monitoring solutions arises from applications in healthcare, environmental sensing, and smart city initiatives among others. The steady increase in both domain-specific and general research outputs reinforces RTM's growing importance in tackling modern challenges and driving technological innovation.

2.3 Statistical Approaches

Statistical techniques have served as the foundation for RTM systems, especially in the contexts of industrial, medical, and engineering applications. These methods utilize probabilistic models and statistical inference to identify anomalies, detect changes, or spot outliers within time-series data streams. One of the main benefits of these approaches lies in their clarity of interpretation, strong theoretical support, and their relatively modest computational demands. Furthermore, they offer a dependable framework for monitoring in situations where clear interpretability, transparency, and formal validation are vital.

Nonetheless, these methods generally presume particular data distributions, are vulnerable to inaccuracies if the model specifications are flawed, and may struggle to adapt or scale with the growing complexity and volume of modern data streams. To overcome these limitations, hybrid approaches that merge the rigorousness of statistical methods with the adaptability offered by ML have gained popularity. Such hybrid methods are especially valuable in environments with shifting data distributions and scarce labeled data.

2.3.1 Control Charts

Control charts, introduced by Walter A. Shewhart during the 1920s, are among the earliest and most fundamental statistical instruments created to support quality control and process supervision [41, 42]. These charts provide a graphical representation that tracks process measurements over time, enabling practitioners to judge whether a process operates within defined limits or has shifted due to identifiable causes [43].

Fundamentally, control charts (refer to Figure 2.4) display observed data points together with control thresholds calculated from historic process behavior. These thresholds commonly correspond to ranges determined by standard deviations around the process mean, marking what is considered normal variation. If the process data crosses these control limits, it often signals an abnormal condition or a change in the process, triggering alerts or prompting further examination. Various types of control charts exist, designed to accommodate different monitoring scenarios including both single-variable (univariate) and multiple-variable (multivariate) processes.

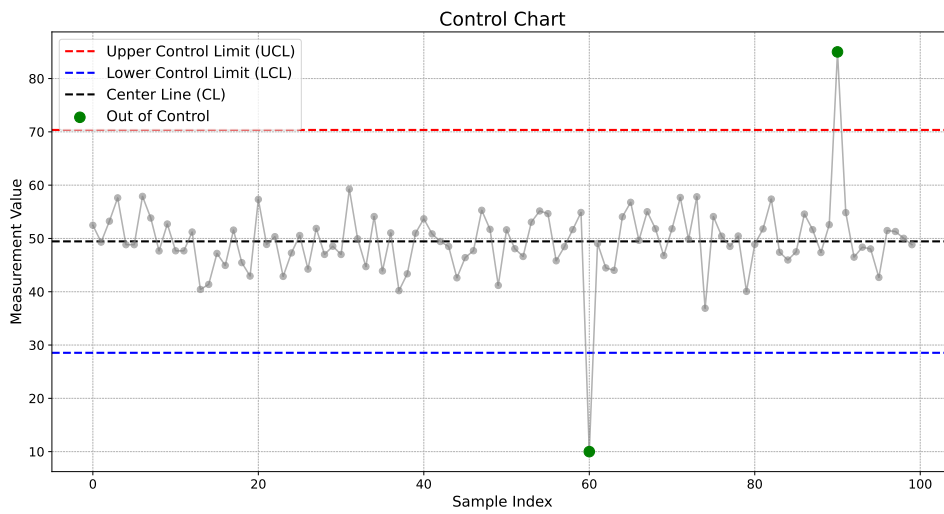
Fig. 2.4 An example of Control Chart ($\bar{X}$)

Table 2.2 presents an overview of the main control chart types deployed in RTM contexts, describing their ability to detect various forms of process change, their operational benefits and constraints, and characteristic industrial settings where each approach has demonstrated particular value.

Table 2.2 Contribution of Control Charts for RTM

Method	Shift Type	Strengths	Limitations	Example
Shewhart	Large, sudden	Simple	Small shift blind	Pharmaceutical batch monitoring
CUSUM	Small, sustained	Drift sensitive	Computation heavy	Chemical process mean shifts
EWMA	Small, gradual	Noise robust	Memory intensive	Semiconductor wafer production
T^2	Multi-sudden	Correlation aware	Outlier sensitive	Tennessee Eastman process
MCUSUM	Multi-sustained	Subtle drift	Complex setup	Automotive assembly lines
MEWMA	Multi-gradual	Adaptive	Stationarity assumption	Petrochemical plant monitoring
Regression-Adjusted	Multi-correlated	High correlation handling	Covariance estimation sensitivity	Continuous casting, multi-stage reactors

Univariate Control Charts

Univariate control charts focus on monitoring a single process variable over time. The principal variants include the Shewhart chart, the CUSUM chart, and the EWMA chart.

Shewhart Chart First introduced by Walter A. Shewhart in 1924, the Shewhart chart is a fundamental instrument in statistical quality control used to assess process stability and variability [44]. It features a central line representing the mean of the process and Upper and Lower Control Limits (UCL and LCL) that define the acceptable range of variation [45]. Observations that fall outside these limits signal special-cause variations warranting further examination [46]. The chart operates under the assumption that the process is stable and influenced only by common-cause variation, flagging deviations beyond control limits as anomalies [47]. Its advantages include simplicity and intuitive visualization of process behavior [48]. However, it is less efficient in detecting slight shifts in the process mean or variance, especially with smaller sample sizes [49]. To improve sensitivity, it is often combined with CUSUM or EWMA charts [50]. The Shewhart chart is widely used across various sectors such as manufacturing and healthcare, playing a critical role in sustaining quality and detecting abnormalities in near real time [51].

CUSUM Chart Developed by E.S. Page in 1954, monitors the CUSUM of deviations from a target value, providing enhanced sensitivity to small process shifts compared to the Shewhart chart [52]. It is especially effective for applications where minor deviations have significant impact, including semiconductor and pharmaceutical manufacturing [45]. The CUSUM statistic aggregates deviations as follows:

$$S_t = \max(0, S_{t-1} + (x_t - \mu) - k)$$

where S_t denotes the cumulative statistic at time t , x_t is the observed measurement, μ is the target mean, and k is a reference or slack value. An alarm is raised when S_t exceeds the predefined control limit H [50]. The heightened sensitivity facilitates early intervention [53], though it may induce false alarms in noisy data. Proper calibration of k and H is critical for effective performance [50]. The CUSUM chart finds practical use in healthcare and manufacturing for detecting subtle changes that affect quality and is typically used together with Shewhart charts for comprehensive monitoring [45].

EWMA Chart Designed to detect small and gradual deviations in the process mean more effectively than traditional Shewhart charts, the EWMA chart assigns progressively greater weight to recent data points, producing a smoothed estimate of the process level [54]. The EWMA statistic at time t is formalized as:

$$\mathbf{Z}_t = \lambda \mathbf{x}_t + (1 - \lambda) \mathbf{Z}_{t-1}$$

where the smoothing constant λ satisfies $0 < \lambda \leq 1$, $\mathbf{x}_t$ is the current observation, and $\mathbf{Z}_{t-1}$ is the preceding EWMA statistic [55]. Control limits are derived from the expected mean and variance of $\mathbf{Z}_t$, assuming the data approximately follows a normal distribution [56]. The EWMA chart excels at detecting subtle process changes [57] and is highly favored in precision industries like semiconductor and chemical manufacturing [45]. Selecting an appropriate λ is vital: lower values increase stability but reduce responsiveness, whereas higher values enhance sensitivity at the risk of more false alarms [49]. When assumptions of normality do not hold, alternate charts like CUSUM or MCUSUM are preferred [50]. Overall, the EWMA chart offers a powerful method for early detection of minor shifts, cementing its role in modern quality assurance [45].

Multivariate Control Charts

Multivariate control charts track multiple interrelated variables simultaneously, facilitating the detection of changes in complex processes that are not evident with univariate charts.

Shewhart Multivariate T^2 -Chart Based on the Mahalanobis distance, monitors the stability of correlated multiple process variables [58]. Its key statistic (for a single observation) is calculated as:

$$T^2 = (\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})$$

where $\mathbf{x}$ represents the observation vector, $\boldsymbol{\mu}$ is the mean vector of the process, and $\boldsymbol{\Sigma}^{-1}$ is the inverse covariance matrix [45]. The control limits derive from the chi-square distribution. The T^2 -chart effectively identifies shifts in mean vectors, proving particularly useful in sectors like semiconductor manufacturing and aerospace production where variables are interdependent [45]. However, it presupposes multivariate normality and may perform poorly when detecting small shifts or with non-normal data; alternative methods like CUSUM or EWMA might be preferred in those scenarios [47],

[50]. Despite its limitations, the T^2 -chart remains a staple tool in multivariate quality monitoring [46].

Multivariate Cumulative Sum (MCUSUM) Chart Extends the CUSUM method to multivariate data by accumulating deviations using the Mahalanobis distance [58, 52]:

$$S_t = \max\left(0, S_{t-1} + (\mathbf{x}_t - \boldsymbol{\mu})^\top \mathbf{S}^{-1}(\mathbf{x}_t - \boldsymbol{\mu}) - h\right)$$

Here, h is the decision limit [45]. An out-of-control signal is issued when the cumulative statistic exceeds this threshold. MCUSUM offers increased sensitivity over the Shewhart T^2 -chart for small changes, permitting earlier detection [50]. However, precise tuning of parameters like h is essential to prevent false negatives or false alarms. The method assumes multivariate normality, continuing to pose a challenge [47]. The MCUSUM chart has applications in sectors including chemical and aerospace manufacturing, where early detection of correlated variable shifts is critical [51].

Multivariate Exponentially Weighted Moving Average (MEWMA) Chart Combines the EWMA concept with multivariate monitoring, enhancing the detection of small and sustained shifts in process means compared to the Shewhart T^2 -chart. It computes:

$$\mathbf{Z}_t = \lambda \mathbf{x}_t + (1 - \lambda) \mathbf{Z}_{t-1}$$

and monitors via:

$$\mathbf{S}_t = \mathbf{Z}_t^\top \boldsymbol{\Sigma}^{-1} \mathbf{Z}_t$$

where $\boldsymbol{\Sigma}^{-1}$ is the inverse covariance matrix. An alert is triggered when $\mathbf{S}_t$ surpasses chi-square-distributed control limits.

Regression-Adjusted Control Charts Regression adjusted control charts tackle the problem of strongly correlated process variables by replacing the original measurements with residuals that more clearly highlight deviations from expected behavior. In the formulation introduced by Hawkins [59, 60], each variable is regressed on the remaining variables, and monitoring is performed on the corresponding residuals, which represent observations after adjusting for their conditional expectation given the others [59]. When these regression based residual charts are used in conjunction with EWMA, they have been shown to outperform traditional T^2 charts for detecting

changes in the covariance structure, especially in settings with very high correlations (above 0.9) [60]. Such methodology has been successfully applied in industrial contexts with strong inter-variable dependence, including continuous casting and complex chemical processing systems. A drawback, however, is the reliance on accurately estimated covariance relationships, which makes the procedure vulnerable to misspecified models when the assumed dependencies do not reflect the true process dynamics. A key restriction of conventional control charting schemes is the implicit assumption that the underlying process is stationary over time, which is often violated in contemporary streaming applications affected by concept drift and shifting distributions. Although more recent developments introduce adaptive limits and kernel based formulations to better accommodate non stationarity, considerable difficulties remain in scaling these methods to large scale sensor networks and IoT environments where both computational demands and latency constraints are severe.

Traditional control charts hinge on the assumption of stationarity, presuming that the underlying process distribution remains constant over time. Modern RTM scenarios, however, involve data that frequently exhibit non-stationarity, high dimensionality, and distributional changes, which challenge conventional chart reliability. To address these issues, recent studies have developed adaptive control charts and ML-assisted approaches that dynamically adjust control limits, enhancing detection accuracy and robustness in complex, evolving systems.

2.3.2 Dimensionality Reduction and Regression Techniques

Handling high-dimensional data often requires dimensionality reduction techniques such as PCA, PLS, and ICA to reduce complexity while preserving essential structural information. These methodologies simplify complex processes by transforming variables into manageable forms for effective monitoring.

Principal Component Analysis Pioneered by Karl Pearson [61], is a key statistical method that reduces correlated, high-dimensional data to a lower-dimensional set of uncorrelated variables called principal components (see Figure 2.5). PCA's strength in process monitoring lies in its ability to capture the majority of data variance within a few principal components, uncovering latent structures, relationships, and trends that might otherwise be obscured [62]. In practice, control charts constructed on principal component scores, such as Hotelling's T^2 and the Squared Prediction Error (SPE), enable early detection of process deviations and potential faults.

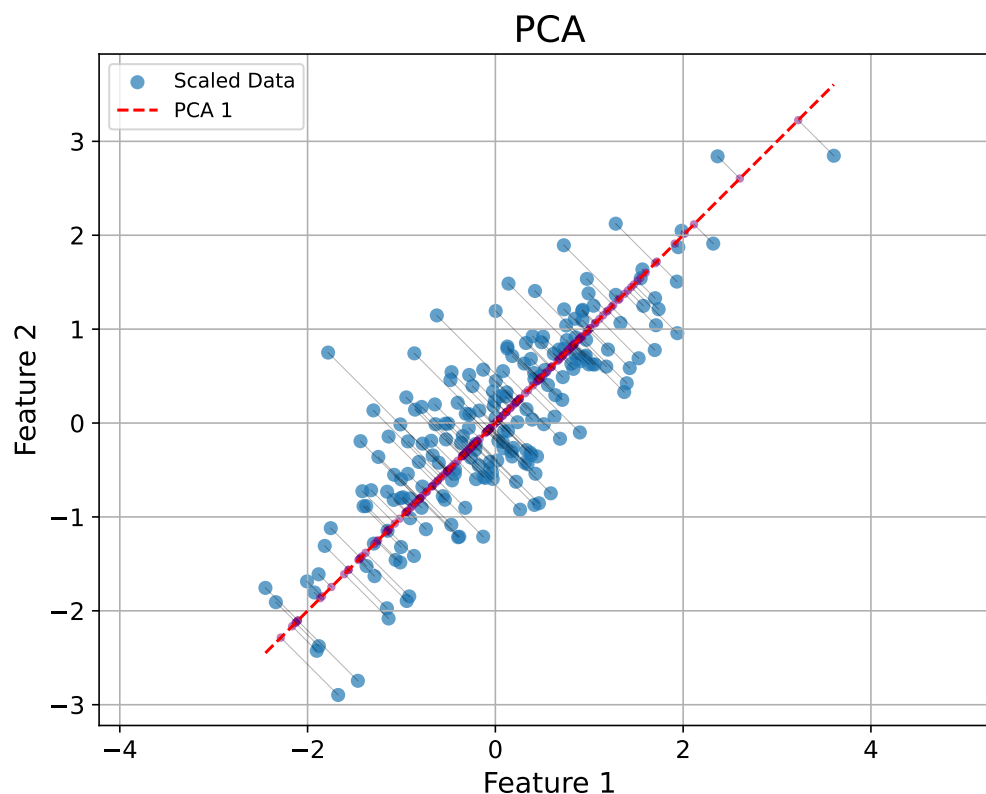


Fig. 2.5 An example of Principal Component Analysis

To improve upon static PCA for RTM, advanced variants like Moving Principal Component Analysis (MPCA) have been developed. MPCA applies PCA on a moving analysis window, updating the principal components as new data arrive [63]. Unlike traditional PCA, MPCA adapts to ongoing process dynamics, effectively tracking abrupt changes in process means or variance, and detecting shifts in correlation structures indicative of subtle anomalies such as sensor drift or gradual equipment degradation common in chemical industries. Another evolution is Multiscale Principal Component Analysis (MSPCA), which integrates wavelet decomposition with PCA to monitor across multiple temporal and frequency scales [64]. By breaking down process signals into components at various resolutions, MSPCA effectively filters noise, mitigates strong autocorrelation, and improves sensitivity to abrupt and gradual faults. Its applications in chemical reactors and power plants have demonstrated superior fault detection performance, aiding predictive maintenance and process optimization [65]. Kernel Principal Component Analysis (KPCA) extends PCA by applying nonlinear kernel functions, enabling the extraction of principal components in complex nonlinear

process data [66]. KPCA is widely used in RTM of batch and continuous processes where interactions between variables may be nonlinear, such as in bioprocessing and advanced materials manufacturing. It maps data into a higher-dimensional feature space to reveal faults and trends invisible to classical linear PCA methods. In addition, contemporary research explores Dynamic Principal Component Analysis (DPCA) [67], Multivariate Image Analysis (MIA) using multiway PCA [68], and sequential PCA methods for batch processes [69], all of which are optimized for handling the high throughput and velocity of data typical in modern industrial settings. For instance, PCA combined with fuzzy logic control has enhanced online fault detection in steel continuous casting operations [70], while multiway PCA supports rapid defect detection in real-time visual inspections on production lines. In summary, PCA and its real-time capable variants (MPCA, MSPCA, KPCA, and MIA) form a potent toolkit for process monitoring and fault diagnosis, effective across chemical, petrochemical, metallurgical, and advanced manufacturing industries. These methods continuously evolve to meet the challenges of high-dimensional, dynamic, and nonlinear data.

Partial Least Squares Developed by Herman Wold, addresses complex multivariate relationships particularly when data exhibit multicollinearity and high dimensionality. Unlike PCA, PLS is supervised and constructs latent variables by maximizing covariance between predictor and response variables. It is highly effective for monitoring regression-based processes, summarizing complex data into components reflecting critical predictor-response relationships [71]. Multiscale PLS (MSPLS) combines wavelet analysis with PLS to model multiscale dynamics, decomposing data via online wavelet transform and using multiblock PLS to capture both global and local features [72]. Kernel PLS (KPLS) further enhances monitoring by capturing nonlinear predictor-response correlations, with modifications allowing high-dimensional feature mapping for improved modeling [73].

Independent Component Analysis Introduced in the early 1990s, decomposes multivariate signals into statistically independent, non-Gaussian source components [74]. Unlike PCA, which targets uncorrelated linear projections, ICA reveals underlying structures by identifying statistical independence, enabling detection of subtle shifts or patterns missed by PCA [75]. ICA finds applications in blind source separation and monitoring of non-Gaussian processes, including industrial quality control.

Table 2.3 provides an overview of the main dimensionality reduction methods used in RTM settings, where each technique is characterized by specific assumptions and strengths that make it appropriate for particular monitoring scenarios.

Table 2.3 Comparison of Dimensionality Reduction Methods for RTM

Method	Assumptions	Strengths	Real-world Example
PCA	Linear, Gaussian	Interpretability, speed	Fault isolation in Tennessee Eastman plant
MPCA/MSPCA	Dynamic processes	Online adaptation	Wastewater treatment monitoring
KPCA	Nonlinear	Captures complex patterns	Nonlinear chemical reactors
PLS	Supervised, collinear	Predictive modeling	Soft sensor development
MSPLS/KPLS	Nonlinear, multiscale	Handles dynamics	Pulp and paper processes
ICA	Non-Gaussian	Blind source separation	Motor fault diagnosis

2.4 Machine Learning Approaches

The combination of statistical techniques with ML methods (SML) has become a cornerstone in the evolution of RTM systems across diverse areas. Classical statistical tools such as control charts, sequential hypothesis testing, Kalman filters, and Bayesian inference have traditionally been employed to detect anomalies, estimate system states, and capture temporal dependencies in streaming data contexts [36, 37]. These methods provide strong theoretical foundations and are effective when the data distributions and system dynamics are well characterized. However, the increasing complexity and magnitude of real-time data streams have created a demand for more flexible, data-driven approaches.

To meet these challenges, ML-based techniques have gained wide traction due to their ability to model complex, nonlinear interactions in data dynamically. Online learning algorithms, including Stochastic Gradient Descent (SGD) and incremental adaptations of SVMs and Decision Trees (DTs), allow continuous model updating as new information becomes available [38]. DL models, especially Recurrent Neural networks (RNNs) and LSTM networks, are extensively used for time series forecasting [18] and anomaly detection because of their proficiency in capturing long-distance

temporal patterns and dependencies [76]. Ensemble methods and hybrid frameworks that integrate statistical reliability with ML’s adaptability have also shown significant promise in raising detection accuracy and reducing false positive rates in dynamic environments [77]. As RTM applications penetrate mission-critical areas such as predictive maintenance, financial fraud prevention, autonomous vehicles, and environmental monitoring, the confluence of ML and statistical principles is shaping the next generation of intelligent, responsive monitoring systems.

ML has emerged as a pivotal paradigm within modern RTM frameworks, owing to its capacity to directly interpret raw, high-dimensional, and heterogeneous data; adapt continuously to evolving conditions; and provide scalable solutions. Unlike traditional statistical models which often depend on explicit assumptions about data distributions and handcrafted feature engineering, ML models learn intricate features from data itself, rendering them capable of generalizing to unseen or novel scenarios.

A broad spectrum of ML methods for streaming data has been developed, spanning online learning algorithms, attention-based and RNNs, ensemble learners, and hybrid statistical-ML combinations. Contemporary research continues to advance RTM capabilities by integrating ML with adaptive control, signal processing, and statistical inference elements, crafting hybrid systems that achieve superior resilience, interpretability, and operational performance.

2.4.1 Supervised Learning in Streaming Contexts

Supervised learning has historically been a core technique for predictive modeling and classification, where algorithms are developed using labeled datasets to map input features to desired outputs. Within RTM systems, supervised learning plays an important role in anticipating critical occurrences such as system faults, equipment malfunctions, or categorical state assignments. These predictive capabilities facilitate proactive maintenance, timely detection of faults, and operational adjustments that help minimize downtime and enhance process reliability [78].

However, the nature of streaming data in many RTM applications presents challenges for supervised learning. These include constantly shifting data distributions, high-speed data arrival, and delays or insufficiencies in obtaining accurate labels. To sustain prediction accuracy, supervised models often need to update incrementally as each new data point arrives—this paradigm, commonly called online learning, ensures models remain relevant despite changing data-generating mechanisms [78]. Several supervised learning methods have proven effective for real-time data:

Support Vector Machines Originally proposed by Boser, Guyon, and Vapnik [79], SVMs are classifiers designed to maximize the margin between different classes by defining an optimal decision boundary (illustrated in Figure 2.6). Their efficacy in handling high-dimensional feature spaces makes them suitable for fault detection in industrial processes, including abnormal operation identification in chemical plants [80] and diagnosing equipment failures in manufacturing [81]. Recent advances have developed online SVM variants that incrementally update classification boundaries to maintain accuracy under evolving process characteristics [78].

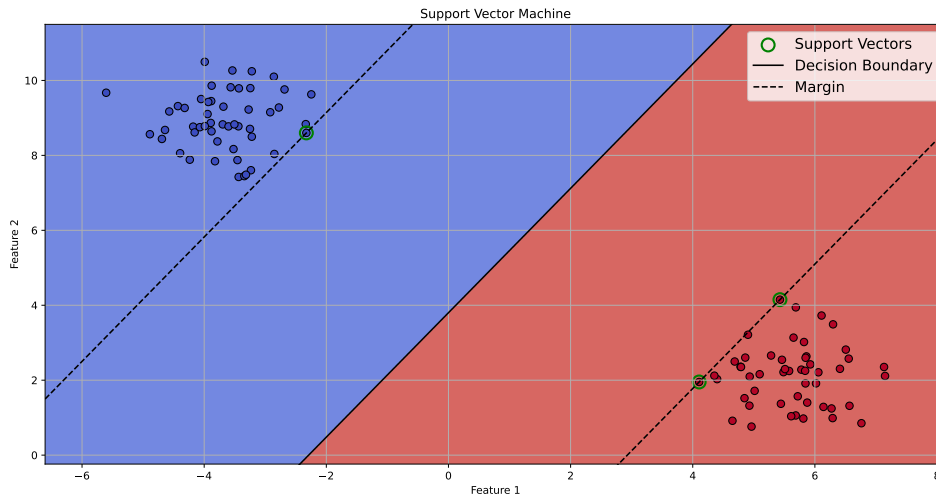


Fig. 2.6 An example of SVMs

k-Nearest Neighbors (kNNs) The kNN algorithm is a simple yet versatile method applicable to classification and regression problems. It determines the label or value of a new instance based on the labels or values of its k nearest neighbors in the feature space (see Figure 2.7), commonly using Euclidean distance as the metric. In RTM applications, kNN has effectively identified faults by comparing current system states against historical data [82]. The algorithm's non-parametric nature allows flexible modeling without explicit training. However, its computational cost can grow significantly with increasing data size, necessitating approximate nearest neighbor searches or data windowing to sustain real-time performance [78].

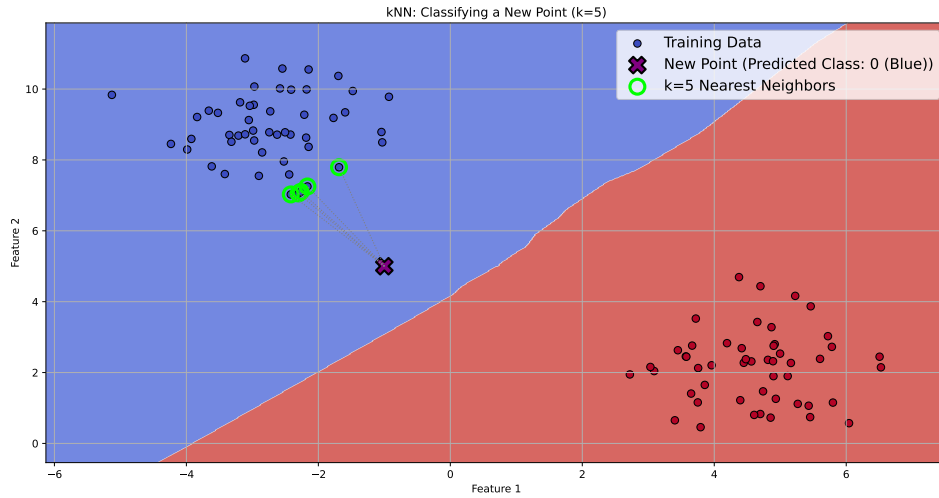


Fig. 2.7 An example of kNN (with $k = 5$ closest counterparts)

Practical Implementations and Challenges In industrial settings, supervised learning models often are linked directly to IoT sensor networks, which continuously collect parameters like temperature, pressure, vibration, and acoustic emissions. For instance, ML analyzes these sensor streams in manufacturing environments to forecast equipment failures and optimize maintenance scheduling. Similarly, supervised models support energy management by predicting load changes and improving efficiency. Despite such progress, several challenges persist, including delayed availability of accurate labels, data imbalance due to rare fault events, and the computational burden of continually retraining models. Moreover, ensuring model interpretability and establishing operator trust remain critical to deploying supervised learning reliably in safety-critical applications. In closing, supervised learning continues to be a vital element within RTM systems by utilizing labeled data to predict impactful events. Its future success depends on adaptive techniques for handling evolving data distributions, computationally efficient algorithms for high-speed streams, and seamless integration with domain knowledge and sensor technologies.

2.4.2 Unsupervised Learning and Anomaly Detection

Unsupervised learning holds significant importance in RTM systems, especially when labeled datasets are scarce or unavailable. This method is particularly effective for anomaly detection by enabling systems to recognize deviations from standard operating behavior without relying on pre-existing examples of failures or irregularities [4]. By

perpetually analyzing incoming data streams, unsupervised techniques maintain a model that represents typical system behavior, thereby proactively identifying unusual patterns that may signal novel faults or process alterations [83].

Clustering-based unsupervised methods provide a powerful strategy for implementing MSPM in the absence of labeled normal and abnormal data [84]. These techniques autonomously discover natural clusters within the process measurements and flag data points that do not conform to these clusters as possible anomalies. For instance, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [85] has been applied successfully in RTM settings [86], as it can identify clusters with arbitrary shapes and classify points outside these clusters as noise or outliers. This characteristic allows DBSCAN to detect previously unseen faults or abnormal conditions without the need for predefined models.

In real-world industrial RTM deployments, unsupervised learning is often combined with sensor networks to continuously monitor equipment health and operational performance [87]. Such techniques enable the automatic detection of subtle changes in parameters like vibration, temperature, or power consumption, allowing predictions of equipment degradation long before failures occur [88]. Moreover, unsupervised methods allow monitoring systems to function effectively in dynamic environments where process conditions shift over time and annotated fault data is either limited or delayed.

Overall, unsupervised learning delivers adaptable and flexible tools crucial for early anomaly detection within RTM frameworks, facilitating timely actions that improve system dependability and reduce unexpected downtime.

2.4.3 Deep Learning for Temporal Sequence Modeling

DL has revolutionized many domains, including computer vision, Natural Language Processing (NLP), and sequence data analysis, due to its ability to autonomously learn hierarchical and sophisticated features from raw data. In RTM, DL architectures have demonstrated strong capabilities for applications including time series forecasting, event and anomaly detection, and predictive maintenance. However, deploying deep artificial neural networks in real-time contexts introduces challenges regarding computational cost, memory consumption, and inference speed.

Various DL architectures and strategies have been adapted to satisfy RTM needs, particularly in streaming data contexts and environments with limited computational resources. While DL offers substantial potential for intelligent RTM, successful application demands a careful trade-off among accuracy, latency, interpretability, and

computational feasibility. Advances in efficient neural network architectures and hardware acceleration are progressively bridging the gap between theoretical models and practical deployment, rendering DL an increasingly integral part of advanced RTM solutions.

Recurrent Neural Networks (RNNs) Including LSTM networks and Gated Recurrent Units (GRUs), are prevalently used to model temporal dependencies within sequential data [76]. These models excel in capturing long-range temporal correlations, making them well suited for RTM tasks such as predicting faults and monitoring system condition. Nonetheless, conventional RNNs require significant computational resources and pose difficulties in parallelization, limiting their applicability in demanding real-time settings.

2.4.4 Reinforcement Learning for Adaptive Monitoring

RL has emerged as a robust framework for decision-making in changing environments. Unlike supervised learning, which depends on fixed labeled data, RL involves an agent interacting with its environment and learning strategies that maximize rewards through trial and error. This approach is especially valuable in RTM where systems must detect anomalies and adapt dynamically to evolving conditions. RL benefits RTM by enabling continuous learning from interactions, handling delayed feedback, and optimizing sequences of actions instead of isolated predictions. Therefore, RL has found applications in a variety of RTM tasks:

Adaptive Thresholding In conventional monitoring, static thresholds often fail to accommodate varying operational conditions. RL methods can dynamically regulate alert thresholds by leveraging historical data and contextual cues, optimizing the balance between false alarms and missed detections [89].

Resource Allocation RTM implementations on edge devices and IoT platforms operate with constrained resources. RL techniques have been utilized to learn efficient policies for allocating sensing, computational, and communication resources to maximize monitoring effectiveness while minimizing energy and bandwidth consumption [90].

Predictive Maintenance In industrial contexts, RL has demonstrated effectiveness in maintenance scheduling by learning from equipment health data and operational

feedback to optimize maintenance timing, aiming to minimize downtime and avoid unnecessary servicing [91].

The fusion of DL with RL, known as Deep RL (DRL), incorporating algorithms like Deep Q-Networks (DQN), Proximal Policy Optimization (PPO), and Actor-Critic models, has broadened RL's applicability in high-dimensional, complex monitoring problems. This integration allows for end-to-end learning from raw sensor inputs, enabling enhanced performance in RTM scenarios.

2.4.5 Online and Offline Learning

Two major learning paradigms exist: offline (batch) learning and online learning. Offline learning assumes availability of the entire training dataset upfront, permitting models to be trained in a static manner, typically requiring significant computational resources and lacking adaptability to changing data streams. Conversely, online learning algorithms update model parameters incrementally as unseen data arrives, foregoing the storage or reprocessing of previous data points [92]. This characteristic makes online learning highly suitable for RTM applications where data streams continuously evolve and distributions shift over time.

Online learning supports rapid model updates with low latency, continuous adaptation, and efficient use of computational and memory resources. These attributes enable RTM systems to detect new patterns or anomalies promptly, ensuring accurate and timely monitoring. Common online learning techniques include Online SGD, which iteratively updates parameters, Passive-Aggressive algorithms known for swift adjustments and noise robustness, and Hoeffding Trees, well-suited for high-velocity data due to their scalable incremental learning [28]. Many of these methods are also equipped to detect and adapt to concept drift, a frequent challenge in real-time environments where data properties change constantly.

Table 2.4 summarizes the role of ML techniques in RTM contexts, outlining their data needs, operational advantages, key drawbacks, and common application areas. This comparison shows that ML frameworks generally surpass purely statistical approaches when confronted with nonlinear relationships and non stationary processes, demonstrating greater flexibility in evolving environments. Nonetheless, important obstacles persist in terms of interpretability, as many ML models lack transparency in their decision mechanisms, and computational overhead, which may hinder responsiveness in environments with strict latency or resource constraints.

Table 2.4 Contribution of ML Methods for RTM

Method	Data	Strengths	Limits	Real-world Example
SVM	Labeled	Robust, accurate	Slow updates	Fault detection
kNN	Labeled	Simple	High memory	Equip. diagnosis
DBSCAN	Unlabeled	Flexible clusters	Density sensitive	Anomaly clustering
LSTM	Sequential	Long dependencies	High latency	Pred. maintenance
RL (DQN/PPO)	Interactive	Adaptive policy	Exploration cost	Threshold tuning
Hoeffding Tree	Streaming	Fast, scalable	Drift lag	Network monitoring

2.5 Conclusions

Traditional MSPM methods remain valuable due to their interpretability and statistical rigor, making them well-suited for structured, stable environments, while ML-based online and streaming approaches offer enhanced flexibility to handle non-stationary and complex data. The combining of these techniques enables robust, accurate, and responsive RTM systems.

Despite these advances, challenges such as improving ML transparency, maintaining robustness against concept drift, and developing scalable solutions for resource-constrained contexts persist. Emerging architectures including edge computing and distributed analytics will play a pivotal role in deploying RTM systems at scale. Continued research integrating MSPM with ML, focusing on efficiency, adaptivity, and explainability, is essential to fully exploit RTM’s potential across industries and disciplines.

2.6 Summary

This chapter has presented a broad overview of RTM systems, emphasizing the integration of MSPM and ML approaches. It covered foundational concepts, the evolution of RTM, key statistical and ML methodologies, and the challenges involved in processing continuous, high-volume data streams in real time. The chapter highlighted diverse application domains, recent advances in algorithmic and system architectures, and the interplay between interpretability, scalability, and adaptivity in RTM solutions.

Chapter 3

Bridging Statistical Process Monitoring and Machine Learning

3.1 Introduction

This chapter builds upon a statistical and machine learning framework, where the integration of classical MSPM methodologies with modern ML techniques was methodically explored. By addressing limitations of traditional MSPM in handling high-dimensional, streaming, and dynamic data settings, this study introduces adaptive frameworks that enhance detection robustness and computational scalability. The following sections develop these concepts further, situating the ensuing discussion within the evolving landscape of real-time anomaly detection.

The accelerated digitization of information and the pervasive use of data acquisition technologies have produced an unparalleled surge in large-scale datasets, collectively termed big data. With continuous technological progress, the scale, diversity, and speed of data generation are expanding, driving the demand for advanced analytical methodologies capable of deriving actionable knowledge from these massive and heterogeneous datasets. While this rapid data proliferation brings vast opportunities, it simultaneously introduces significant analytical and computational challenges.

Within the domain of SPM, the analysis of high-dimensional data streams presents distinct difficulties, primarily due to the “curse of dimensionality”, where data sparsity in multidimensional spaces hinders the detection of relevant structures, deviations, and anomalies. Additionally, as the number of process variables increases, modeling and supervising intricate interdependencies among them become more complex. The challenge intensifies further when MSPM is implemented in real-time environments,

which demand rapid computation and timely decision-making for handling voluminous and fast-moving data [93].

Integrating MSPM with ML methodologies presents a promising pathway toward addressing these complexities. MSPM provides a statistically grounded and interpretable modeling framework, while ML contributes flexible, adaptive mechanisms suitable for nonlinear and evolving systems. Cluster Analysis (CA) stands at the crossroads between these two domains; it originates from statistical theory yet plays a critical role in ML applications. By dividing data into clusters representing normal operational states, clustering facilitates anomaly detection, particularly when fault labels are limited or unavailable [94].

Building upon this foundation, the current study proposes a robust framework derived from CA principles. The proposed method is benchmarked against traditional multivariate control chart techniques, such as the Hotelling T-Square statistic, as well as against clustering validity measures, including Dunn's Index. Results from extensive simulation experiments indicate that the newly developed approach surpasses conventional methods in managing high-dimensional streaming datasets characterized by strong inter-variable correlations.

The proliferation of big data has introduced a new era defined by continuous, high-velocity data streams originating from diverse sources, such as financial markets, industrial sensors, and social media networks [95, 96, 97]. While these data streams enable real-time analysis and swift anomaly identification, they also present technical challenges associated with dynamic data distributions, computational scalability, and the development of adaptive, low-latency algorithms [1].

Anomalies which are data points that substantially deviate from normal behavioral patterns, often convey essential information. For example, anomalies in financial transactions can signify fraudulent activities, while irregular sensor readings in industrial contexts may indicate equipment failures or gradual deterioration. Early and accurate identification of such anomalies facilitates preventive maintenance and risk mitigation [4, 5].

Processing and interpreting continuous data streams effectively require algorithms capable of operating efficiently without depleting computational or memory resources [98]. The increasing dimensionality and intricate correlation architecture in modern datasets emphasize the necessity of hybrid frameworks combining the statistical precision of MSPM with the adaptive flexibility of ML. Clustering-based methods are particularly suited for this integration, as they categorize data by structural similarity without needing labeled fault information [94].

In summary, achieving effective real-time monitoring systems relies on uniting the complementary advantages of SPM and ML. This integrated approach provides the critical foundation necessary for modern industries to address operational complexity, enabling early anomaly detection capabilities, adaptive control strategies, and continuous system enhancement initiatives.

3.2 Motivation

The exponential growth of big data has profoundly transformed how organizations operate, creating vast possibilities while simultaneously introducing complex challenges [99]. As data are being generated at extraordinary scales and speeds from diverse sources such as social media platforms, the IoT, and transactional systems [100], the necessity for effective statistical monitoring mechanisms has become indispensable [101]. This necessity arises from the need to ensure high data quality, enable real-time insight generation, detect emerging patterns and trends [102], and maintain compliance within increasingly intricate digital and regulatory environments.

A central motivation for integrating statistical monitoring within big data contexts lies in the imperative to preserve data quality. Poor data quality undermines analytical reliability and can give rise to erroneous interpretations and detrimental business decisions, potentially incurring substantial financial repercussions [103]. To mitigate such risks, organizations must employ statistical methodologies capable of detecting anomalies, inconsistencies, and outliers that compromise the integrity of their datasets. This is particularly critical in data-sensitive domains such as finance and healthcare [104], where precision and reliability are paramount. In these industries, deploying real-time statistical monitoring facilitates the immediate detection of irregularities, preventing operational disruptions and safeguarding decision-making integrity.

Moreover, the demand for real-time, data-driven decision-making further reinforces the importance of robust statistical monitoring frameworks. In financial institutions, these systems are instrumental in detecting fraudulent transactions and unusual trading behaviors [105], enabling rapid responses to evolving risks. The agility afforded by such monitoring not only improves operational resilience but also enhances competitive advantage through timely strategic adjustments. Similarly, in retail environments, continuous monitoring helps organizations analyze consumer behaviors, optimize inventory control, and deliver personalized customer experiences through adaptive predictive insights.

Statistical monitoring also serves as a crucial tool for uncovering patterns and long-term trends across extensive datasets. Techniques such as time series analysis and control chart methodologies allow organizations to monitor key performance indicators, forecast future trajectories, and refine strategic objectives. For instance, continuous surveillance of medical data can substantially improve patient outcomes by facilitating early recognition of deviations that signal potential clinical deterioration [106]. Key industries that rely heavily on rigorous statistical oversight include finance, healthcare, manufacturing, cybersecurity, and marketing. In finance, real-time monitoring is indispensable for detecting fraud and managing systemic risk [107]; in manufacturing, it supports process optimization and waste minimization [108]; in cybersecurity, it enables real-time network surveillance for rapid threat mitigation [109]; and in marketing, it empowers organizations to monitor consumer sentiment to adapt branding and outreach strategies [110].

An additional driving factor behind the adoption of statistical monitoring relates to compliance with evolving regulatory standards. In sectors such as finance and healthcare, where stringent legal frameworks govern data handling and reporting, statistical monitoring systems facilitate continuous supervision of data processes, flagging potential compliance breaches at early stages [111]. By institutionalizing proactive risk management practices, organizations not only reduce exposure to regulatory penalties but also strengthen stakeholder confidence and enhance reputational capital. Furthermore, the adaptability and scalability of these monitoring frameworks are vital in addressing the continuously changing dynamics of data ecosystems. As underscored by [112], the development of flexible, continuously updating monitoring systems capable of adapting to novel data behaviors and anomalies is key to ensuring organizational responsiveness and long-term sustainability.

Nevertheless, traditional statistical methods face considerable limitations in managing the immense volumes, high dimensionalities, and rapid rates of modern data streams [113]. Monitoring high-frequency, multi-dimensional datasets, especially those characterized by continuous inflow, presents nontrivial analytical and computational challenges. A prominent obstacle is the curse of dimensionality, wherein the exponential expansion of the data space hampers the identification of significant patterns and relationships [114]. This problem frequently results in model overfitting [115], whereby algorithms perform admirably on training data yet fail to generalize to unseen cases. While smaller datasets may mitigate this issue by fostering simpler and more generalizable models, the computational intensity of large data environments has propelled the

advancement of distributed and parallel stream processing frameworks designed for adaptive data handling [116].

The effective examination of continuous data streams necessitates real-time analytical methodologies capable of promptly detecting distributional shifts or system anomalies. Conventional batch-processing techniques are inadequate for such dynamic tasks. In response, the concept of rational subgrouping offers an effective alternative. By segmenting data streams into smaller, thematically or temporally relevant subgroups, organizations can enhance monitoring precision and computational efficiency. For instance, sensor data may be categorized according to device type, operational zone, or time interval, allowing focused analysis on the most pertinent subsets. This targeted approach enables the early detection of behavioral changes, clearer identification of emerging trends, and faster decision-making without overwhelming computational resources. Rational subgrouping thus improves both the speed and accuracy of real-time monitoring by optimizing processing focus and minimizing redundant effort.

Moreover, high-frequency data streams often contain substantial noise components [117], which obscure latent patterns and can trigger false alarms during anomaly detection. The computational complexity associated with processing such high-dimensional data compounds these difficulties, demanding substantial resource allocation and extended processing times that may render real-time analytics impractical. Beyond computational constraints, interpretability emerges as a critical issue. Complex dependency structures within high-dimensional data often remain only partially understood, challenging the derivation of reliable, actionable insights. This lack of interpretability may compromise the robustness and scalability of monitoring systems as data environments continue to evolve.

The challenges outlined above highlight the pressing necessity for advanced techniques capable of efficiently monitoring high-dimensional, fast-evolving data streams. As the volume, variety, and velocity of data continue to expand, traditional approaches are proving insufficient to meet the demands of contemporary big data ecosystems. This situation underlines the importance of developing innovative statistical and computational frameworks that combine scalability, adaptability, and precision to handle the dual complexities of dimensionality and streaming velocity effectively.

3.3 Established Multivariate Monitoring Techniques

MSPM is extensively employed for the supervision and regulation of processes characterized by multiple correlated variables. MSPM frameworks commonly utilize control

charts, which generalize traditional univariate charts to accommodate several variables simultaneously, as well as latent variable techniques such as PCA to capture intricate inter-variable dependencies and to interpret out-of-control conditions [42]. Within this methodological spectrum, Hotelling's T^2 statistic stands as one of the most fundamental tools, providing a statistical basis for detecting shifts in process means and variances. Despite its widespread adoption and robust theoretical foundation, the effectiveness of Hotelling's T^2 diminishes when applied to highly complex processes, particularly in cases where the underlying data deviate from the multivariate normality assumption.

The foundational principles and practical applications of control charts have been elaborated in Chapter 1, supplemented by illustrative examples such as Figure 2.4. Building upon that foundation, the current section focuses on the pivotal role that control charts play within MSPM systems, highlighting their capability to detect process instabilities and deviations when managing modern, data-rich industrial environments.

3.3.1 Control Charting

Control charts, as outlined earlier, constitute a fundamental component of SPM, designed to identify temporal variations that may signal deviations from process stability. Variations within a process are broadly classified into two categories: common cause variation, which represents inherent and expected fluctuations in a stable process, and special cause variation, which indicates the presence of irregular or assignable factors that require prompt investigation and corrective intervention [118, 119]. The ability to effectively distinguish between these sources of variation is crucial for determining whether a process remains in a state of statistical control or if it has experienced a shift that necessitates adjustment.

Beyond the classical Shewhart, CUSUM, and EWMA charts, numerous other control chart varieties have been developed to address diverse data types and analysis objectives. The X-bar chart, for instance, monitors sample means, with control limits statistically derived to highlight significant deviations from established targets [47, 120]. Variants such as R charts are used to assess sample range and variability, while attribute-based charts such as the P, NP, C, and U charts focus on evaluating defect proportions, counts, and occurrences. Collectively, these tools provide a comprehensive suite for quality assurance and process stability evaluation across both continuous and discrete manufacturing and service contexts.

Control charts offer a strategic advantage by detecting emerging trends, process drifts, and cyclical patterns before they escalate into full-scale system failures or production defects. Their diagnostic capability supports early warning mechanisms,

enabling organizations to implement preventive maintenance, minimize downtime, and enhance production efficiency. By identifying assignable causes of variability, these methods serve as essential tools for root cause analysis, driving continuous improvement and cost reduction across various industrial sectors, including manufacturing, energy, and healthcare.

Despite their proven utility, traditional control charting techniques encounter substantial limitations when applied to high-velocity, high-dimensional, and dynamic data streams characteristic of modern industrial and IoT-driven environments. Conventional approaches often rely on static thresholds and offline data aggregation, rendering them less responsive to rapidly evolving systems [121, 122]. To address these challenges, advanced RTM systems integrate control charts with data preprocessing steps (such as cleaning, normalization, and feature extraction) to ensure adaptability and improved performance in streaming contexts. Such preprocessing is essential to handle nonstationary data behavior, reduce noise, and maintain sensitivity to process shifts [123].

Recent advancements have also extended traditional control charting into adaptive and intelligent frameworks that exploit the capabilities of ML and artificial intelligence for dynamic decision-making. These systems can self-calibrate control limits, recognize evolving patterns, and adjust parameters in response to contextual shifts. For instance, in smart manufacturing settings, adaptive control charts combined with sensor fusion strategies can analyze data streams from multiple equipment units in real time, identifying subtle performance drifts or early signs of system degradation. IoT-enabled control charts can integrate with predictive maintenance algorithms, triggering automated alerts when monitored parameters exceed learned thresholds, thereby facilitating a proactive approach to process management.

In high-dimensional applications such as semiconductor production, healthcare monitoring, or energy grid management, hybrid control systems that blend statistical techniques with ML-based anomaly detection have demonstrated superior adaptability. These systems employ clustering, PCA, or neural network-based feature extraction methods to reduce dimensionality and isolate essential process characteristics. The integration of such intelligent monitoring frameworks allows organizations to transition from retrospective fault detection to predictive, condition-based control strategies, supporting greater agility and resilience within data-intensive operational environments.

In summary, while control charting remains one of the most established methods in SPM, its evolution towards adaptive, data-driven implementations has transformed it into a powerful framework for real-time quality assurance. The convergence of

traditional statistical control and ML-based signal analysis ensures that modern control charts remain relevant and effective in addressing the complex challenges posed by continuous, high-dimensional big data streams.

Hotelling's T-Square Control Chart

The Hotelling's T-squared statistic, denoted as T^2 , is a foundational measure in multivariate statistical analysis and has long been employed in both one-sample and two-sample hypothesis testing scenarios [124]. Although introduced in the early 1930s, the statistic remains highly relevant, particularly for modern applications involving multivariate and high-dimensional process data. The T^2 statistic quantifies the separation between the sample mean vector, $\bar{x}$, and the hypothesized population mean vector, $\boldsymbol{\mu}$, incorporating the covariance structure $\boldsymbol{\Sigma}$ within a sample size of n . The general formula is expressed as:

$$T^2 = n(\bar{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\bar{x} - \boldsymbol{\mu})$$

Under the null hypothesis $H_0 : \bar{x} = \boldsymbol{\mu}$, and assuming the variable vector of dimension p follows a Multivariate Normal Distribution (MND), the T^2 statistic follows a scaled F -distribution given by:

$$T^2 \sim \frac{p(n-1)}{n-p} F_{p,n-p}$$

where $F_{p,n-p}$ denotes the F -distribution with p and $n-p$ degrees of freedom. The critical values derived from this distribution define the control limits for the Hotelling's T^2 chart, depending on the selected significance level α :

$$\text{UCL} = \frac{p(n-1)(n+1)}{n(n-p)} F_{\alpha,p,n-p}$$

and the LCL corresponds to zero, as negative T^2 values are not meaningful in this context [125].

Figure 3.1 illustrates the distribution of observations in a two-dimensional space. Here, purple points correspond to in-control reference samples, while black points indicate potential out-of-control conditions. The bold blue point denotes the mean of the in-control group, contrasted with the bold red point representing the out-of-control group mean. The dashed green ellipse captures the boundary defined by the 95th percentile UCL.

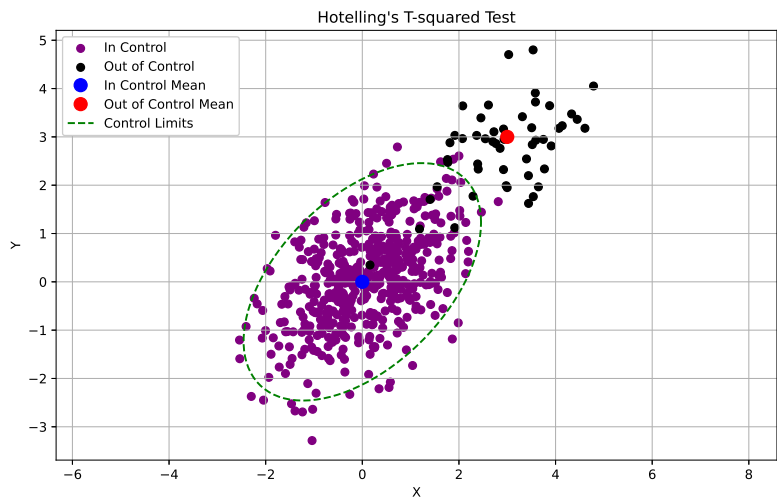


Fig. 3.1 Scatter plot of Hotelling’s T^2 statistic in two-dimensional space showing in-control (purple) and out-of-control (black) points. The dashed green ellipse depicts the UCL.

Similarly, Figure 3.2 presents a temporal control chart representation of Hotelling’s T^2 statistic, where data points exceeding control limits (illustrated as the black cluster) indicate deviations from normal process operation.

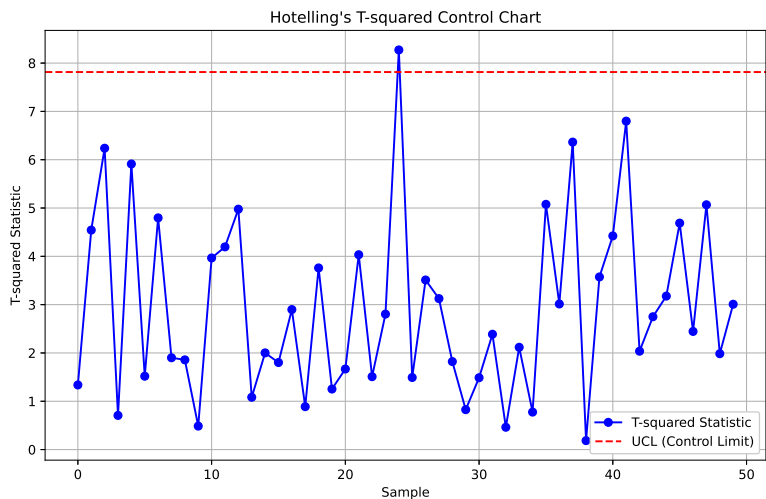


Fig. 3.2 Control chart of Hotelling’s T^2 statistic across time. Values surpassing the control limits signify process deviations requiring investigation.

The Hotelling’s T^2 statistic serves as a powerful tool for identifying multivariate anomalies and shifts, and its applicability spans multiple scientific domains. In

finance [126] and psychology [127], T^2 enables the detection of structural changes and group differences, while in bioinformatics [128], it supports hypothesis testing concerning population mean vectors. Industrially, it assists in real-time analysis of sensor arrays to identify abnormal operations [129], and in cybersecurity, it facilitates the detection of network intrusions through anomalous traffic activity patterns [130].

Yet, the accuracy of the Hotelling's T^2 test crucially depends on the assumption of multivariate normality. Significant deviations from normality, such as skewed or heavy-tailed data distributions, may lead to inflated false-alarm rates or weakened detection power [50, 131, 132]. Furthermore, as data dimensionality increases, the chi-square approximation underlying the T^2 test becomes less accurate, and the sample covariance matrix $\mathbf{\Sigma}$ tends to become ill-conditioned or singular [133]. These challenges reduce the robustness of the standard T^2 method when monitoring high-dimensional datasets.

Adjusted and High-Dimensional Extensions For small-sample or high-dimensional contexts where p approaches or exceeds n , several regularization strategies have been introduced to improve performance. One common adjustment is the use of a shrinkage estimator for the covariance matrix, formulated as:

$$\mathbf{\Sigma}_\lambda = (1 - \lambda)\mathbf{S} + \lambda\mathbf{I}$$

where $\mathbf{S}$ denotes the sample covariance matrix, $\mathbf{I}$ is the identity matrix, and λ is a shrinkage parameter controlling the trade-off between sample-based and regularized covariance estimation. This adjustment improves numerical stability and mitigates the effect of multicollinearity in high-dimensional datasets.

Alternatively, when the dimensionality is extreme, PCA can be integrated into the Hotelling framework by projecting the original data onto a lower-dimensional subspace containing the most significant variance directions. The T^2 statistic in this reduced subspace is then computed as:

$$T^2 = \mathbf{t}^T \mathbf{\Lambda}^{-1} \mathbf{t}$$

where $\mathbf{t}$ represents the principal component score vector and $\mathbf{\Lambda}$ the diagonal matrix of retained eigenvalues. This hybrid approach, often called the Hotelling–PCA chart, preserves detection sensitivity while avoiding computational instability in high- p environments.

Overall, the Hotelling's T^2 control chart remains an essential methodology for multivariate process monitoring. Modern variants continue to adapt its robust theoretical base for contemporary challenges such as streaming analytics, high-dimensional data, and dynamic process environments, ensuring it remains an integral tool for multivariate quality and anomaly detection in both industrial and data-intensive research domains.

3.3.2 Estimating the In-Control Distribution

The effective use of Hotelling's T^2 control charts depends on accurate knowledge of the in-control process distribution. Specifically, the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. In most practical applications, these parameters are unknown and must be estimated from empirical data. The standard procedure to obtain such estimates is carried out during Phase I of process monitoring, where historical in-control observations are collected to construct a statistical baseline. This baseline distribution is subsequently used to establish control limits, such as the UCL.

The Phase I procedure typically commences by accumulating a sufficiently large dataset of samples believed to be representative of in-control operation. For each sample i , the corresponding feature vector $\mathbf{x}_i$ is computed and used to estimate both the process mean vector and the variance-covariance matrix as follows:

$$\bar{\mathbf{x}} = \frac{1}{m} \sum_{i=1}^m \mathbf{x}_i, \quad \mathbf{S} = \frac{1}{m-1} \sum_{i=1}^m (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T$$

where m represents the total number of in-control observations. Using these estimates, the test statistic for each in-control sample is computed via:

$$T_i^2 = (\mathbf{x}_i - \bar{\mathbf{x}})^T \mathbf{S}^{-1} (\mathbf{x}_i - \bar{\mathbf{x}})$$

The resulting collection of T_i^2 values forms the empirical in-control distribution of the statistic. From this set, a chosen upper-tail percentile corresponding to the desired significance level α (e.g., the 95th percentile) is identified to serve as the control threshold:

$$\text{UCL} = T_{(1-\alpha)}^2$$

A value of $T_i^2 > \text{UCL}$ indicates a potential out-of-control condition and warrants further examination of the process state.

It is important to emphasize that this approach assumes the stability of the underlying in-control process. Consequently, while effective for abrupt or clearly defined

process shifts, it might lack sensitivity to gradual drifts, temporal dependencies, or complex nonlinear variations. Furthermore, Phase I estimation requires a considerable number of representative samples to minimize estimation error, as inaccurate estimates of $\boldsymbol{\mu}$ or $\boldsymbol{\Sigma}$ can markedly degrade chart performance.

To address these limitations, modern extensions of Phase I procedures incorporate adaptive mechanisms. Examples include recursive estimation of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ through exponential forgetting factors or Bayesian updating, allowing the in-control model to adjust dynamically as new data become available. Such approaches reduce the need for exhaustive initial sampling and improve robustness in environments where process characteristics evolve gradually over time.

Ultimately, the accuracy and reliability of the in-control distribution estimation directly determine the sensitivity and false-alarm properties of Hotelling's T^2 charts. Therefore, the design of Phase I must carefully balance the trade-off between historical sample size, responsiveness to process changes, and computational efficiency.

Example of UCL Estimation Consider a process with a true mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ in dimension $p = 3$. Suppose the Phase I sample size m varies from 20 to 200 in increments of 20, with independent in-control observations drawn from a MND $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. For each sample size, the empirical distribution of T^2 statistics is computed, and the 95th percentile is used as the UCL estimate.

The following table illustrates how estimated UCL values converge toward the theoretical value as sample size increases:

Phase I Sample Size, m	Estimated UCL (95th Percentile)
20	9.45
40	8.12
60	7.68
80	7.54
100	7.49
120	7.46
160	7.44
200	7.43

The theoretical UCL for $p = 3$ and large m at the 95% confidence level is approximately 7.43, demonstrating that larger Phase I samples yield more precise estimates and thus enhance control chart reliability.

This example underscores the importance of collecting adequate Phase I data for reliable threshold estimation, which critically influences the chart's ability to detect true process shifts while controlling false alarms.

3.4 Clustering-Based Charts for Data Stream Monitoring

Traditional process monitoring frameworks predominantly rely on well-established metrics such as the T^2 statistic, which has been extensively employed to identify deviations in multivariate processes. In this section, an alternative monitoring method is introduced that integrates CA principles with conventional SPM methodologies. This approach harnesses ML clustering techniques to improve the detection of process anomalies within multivariate operational contexts. Unlike traditional control charts that depend on predefined statistical assumptions such as normality, cluster-based control charts capitalize on the inherent grouping structure present in the data. By recognizing clusters that correspond to natural groupings or distinct operational regimes, these methods enable the monitoring of deviations not only within individual clusters but also across clusters. This capability proves particularly advantageous in complex processes where multivariate data exhibit nonlinear dependencies, non-Gaussian distributions, or other irregular behavioral patterns that render standard control charting techniques less effective.

Cluster-based control charts operate by treating sequential data samples as distinct clusters and employing appropriate cluster separability indices to assess process stability. In the current section, a recently proposed monitoring scheme is elaborated, wherein successive data samples are interpreted as individual clusters. Subsequently, Dunn's Index (DI) alongside a collection of recently developed cluster separability metrics are utilized to evaluate clustering quality and detect anomalous process states. The application of these indices provides a quantitative basis for distinguishing between normal process variability and genuine shifts indicative of process deterioration or faults.

Finally, practical considerations regarding the deployment of these recently proposed control charts in high-frequency data stream environments are discussed. The computational efficiency, scalability, and adaptability of clustering algorithms and separability measures become critical factors when implementing RTM systems. Efficient incremental clustering methods and online computation of validity indices facilitate timely anomaly detection without imposing prohibitive computational bur-

dens, thereby enabling effective process surveillance in modern data-intensive industrial and Cyber-Physical Systems (CPS).

3.4.1 Creating Rational Subgroups in Data Streams

Data streams represent a continuous, high-velocity flow of information originating from a wide array of sources, including industrial sensors, IoT devices, social media platforms, financial transaction systems, and web clickstream analytics [134]. Unlike static datasets, which are collected, stored, and subsequently processed in batches, data streams are inherently dynamic and necessitate real-time or near-real-time analytical procedures. This continuous influx of observations captures the temporal evolution and fluctuating characteristics of underlying processes, rendering data streams indispensable for applications demanding prompt responsiveness to emergent changes, anomalies, or system faults.

Within the framework of SPM schemes, the perpetual and high-frequency nature of data streams necessitates the establishment of rational subgroups to partition the flow into analytically tractable units. Rational subgrouping serves as a fundamental principle ensuring that monitoring systems effectively capture transient variations while simultaneously preserving the overarching temporal dynamics of the process being observed. These subgroups are constructed by segmenting the data stream into sequential samples, which is essential for the meaningful application of cluster separability indices, such as Dunn’s Index, which inherently require the presence of at least two distinct clusters to yield interpretable evaluations.

To achieve an optimal balance between computational efficiency and the capacity to detect genuine process variability, it is advisable to maintain relatively small subgroup sizes. A minimum subgroup size of approximately ten observations is commonly recommended [135], as this threshold not only facilitates expedient computation of separability metrics but also ensures that the data stream can be analyzed with minimal latency. Smaller subgroup dimensions reduce the computational burden and memory requirements, thereby enhancing scalability and enabling sustained RTM even in environments characterized by extremely high data velocities.

Figure 3.3 provides a visual illustration of rational subgroups, each containing ten observations, generated from a bivariate normal distribution. This figure demonstrates how consecutive subgroups are delineated within a streaming context. Specifically, the blue points represent sample S_n , whereas the orange points correspond to the immediately succeeding sample S_{n+1} . Such visual depictions clarify how sequential

samples can be viewed as evolving clusters, forming the basis for continuous monitoring and anomaly detection through the computation of separability indices over time.

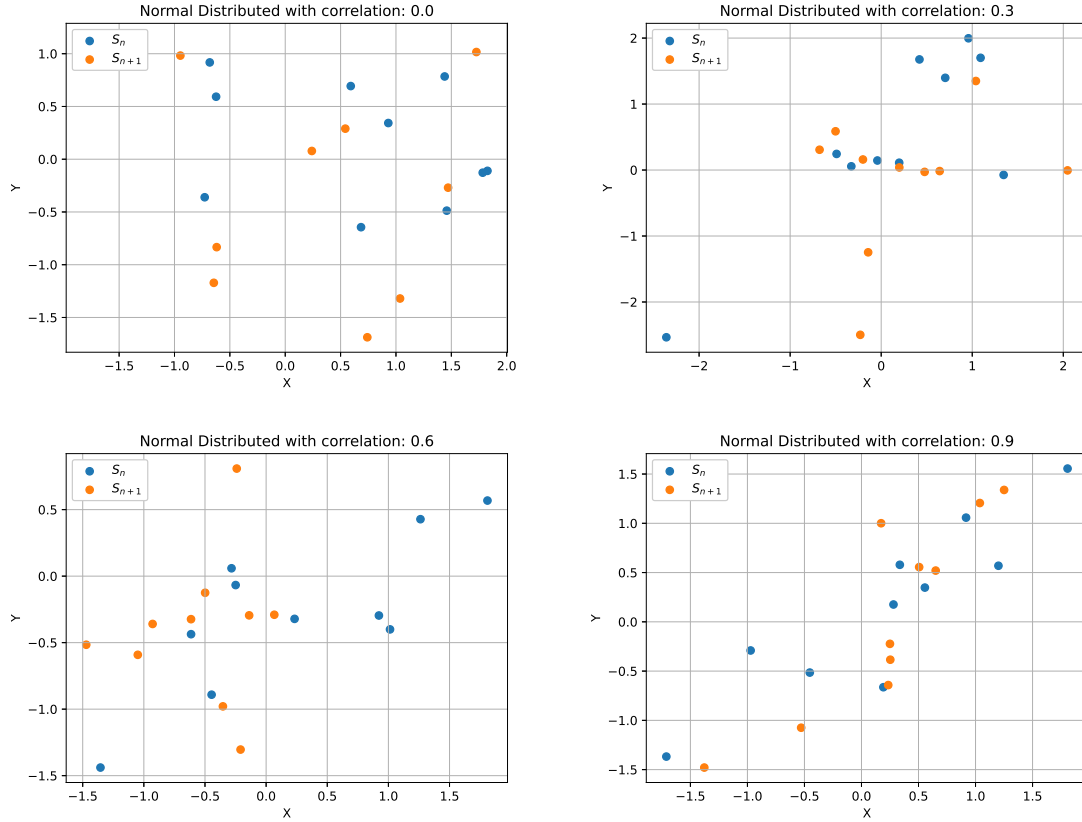


Fig. 3.3 Sequential samples extracted from a data stream following a MND. The blue points represent the S_n sample, while the orange points correspond to the subsequent S_{n+1} sample.

By sequentially partitioning data streams into such rational subgroups, monitoring frameworks can efficiently leverage cluster-based metrics to track process stability, detect shifts in operational regimes, and identify anomalous behaviors in a computationally sustainable manner suitable for modern high-frequency industrial and digital environments.

3.4.2 A Cluster Analysis Approach for Robust Data Stream Monitoring

The following monitoring scheme presents a recently introduced methodology that evaluates process behavior by treating sequential samples as distinct clusters and

monitoring their separability through appropriately defined distance-based indices. This framework offers a direct approach to process monitoring by focusing explicitly on the spatial relationships between successive samples. In contrast to traditional control charts that often rely on parametric assumptions or projections onto lower-dimensional spaces, this scheme leverages raw distance metrics to provide a straightforward and interpretable means of assessing process stability. Within this framework, each sequential sample collected from the data stream is conceptualized as an individual cluster, and the degree of separability between these clusters is quantified using various distance-based measures. For instance, metrics such as the maximum pairwise distance, mean distance, and standard deviation of distances between observations in consecutive samples are employed to evaluate the spatial configuration and dispersion characteristics. These distance-based statistics are fundamental for constructing separability indices, as they encapsulate both the internal cohesion within clusters and the external separation between them.

During Phase I implementation, these indices are sequentially calculated using a historical dataset composed exclusively of in-control samples, thereby establishing a statistical baseline representative of normal process behavior. For each pair of consecutive samples within the Phase I dataset, the relevant distance metrics are computed and aggregated to construct an empirical distribution of separability values. This distribution serves as the foundation for determining control limits that reflect the expected range of separability under stable operating conditions. Based on the desired sensitivity for detecting out-of-control conditions, control limits are established by selecting a specific upper-tail percentile (for example, the 95th percentile) from this empirical distribution. These limits subsequently function as thresholds during Phase II monitoring, where newly acquired sequential samples are continuously evaluated. Any separability index value falling outside the established control limits signals a potential deviation from in-control behavior, prompting further investigation or corrective action.

Consequently, the utilization of sequential samples in conjunction with distance-based separability indices provides a robust, computationally efficient, and interpretable foundation for real-time process monitoring. A critical aspect of this approach lies in the selection of an appropriately defined separability index. Such an index must not only reveal meaningful insights into the cohesion within samples and the separation between them but also exhibit sufficient sensitivity to detect subtle shifts or gradual drifts in process dynamics. The subsequent sections delve into specific separability indices, including the well-established DI, and introduce a recently developed family of indices specifically designed to enhance the detection of evolving clustering patterns

and anomalies in sequential data streams. These indices aim to improve monitoring performance by incorporating refined distance measures and adaptive mechanisms suited to the unique challenges posed by high-frequency, high-dimensional streaming environments.

Dunn Index

The DI serves as a valuable metric for evaluating the quality of cluster separation by leveraging the fundamental concepts of inter-cluster and intra-cluster distances [10]. The minimum inter-cluster distance is defined as the smallest distance observed between any two data points residing in different clusters, thereby quantifying the degree of separation between clusters. Conversely, the maximum intra-cluster distance represents the largest pairwise distance within a single cluster, effectively characterizing the internal dispersion or diameter of that cluster. The DI is designed to maximize inter-cluster separation while simultaneously minimizing intra-cluster diameter, thus enhancing overall clustering quality and interpretability. This fundamental relationship is visually illustrated in Figure 3.4, where both the inter-cluster and intra-cluster distances are depicted to emphasize the optimization objectives inherent in the DI formulation.

When applied to scenarios involving intersecting or overlapping clusters, the DI exhibits notable limitations, primarily due to its dependence on clear separability and compact cluster structures. In situations where clusters overlap significantly or exhibit irregular boundaries, the index may produce low values, even when the clustering solution is contextually meaningful or reflects genuine underlying data patterns. Consequently, in such cases, it becomes necessary to employ alternative clustering evaluation metrics or hybrid methodologies to obtain a more accurate and nuanced assessment of clustering quality.

The computation of the DI relies on calculating the pairwise Euclidean distance between all points in two distinct clusters, denoted as C_1 and C_2 . Let $\mathbf{a}_i \in C_1$ and $\mathbf{b}_j \in C_2$ represent individual observations within each cluster. The Euclidean distance between each pair of points $(\mathbf{a}_i, \mathbf{b}_j)$ from clusters C_1 and C_2 , respectively, in a p -dimensional feature space is formally given by:

$$d_{ij} = \sqrt{\sum_{k=1}^p (a_{ik} - b_{jk})^2}$$

where a_{ik} and b_{jk} denote the coordinates of points $\mathbf{a}_i$ and $\mathbf{b}_j$ along the k -th dimension. This distance calculation is performed for every pair of points $(\mathbf{a}_i, \mathbf{b}_j)$, with $i =$

$1, 2, \dots, n_1$ and $j = 1, 2, \dots, n_2$, where n_1 and n_2 represent the sizes of clusters C_1 and C_2 , respectively.

In light of the preceding considerations, the DI is formally calculated using the following expression:

$$DI = \frac{\min_{i,j} d_{ij}}{\max_n d_{\max}(C_n)},$$

where $d_{\max}(C_n)$ denotes the maximum intra-cluster distance within the n -th cluster. The graphical representation provided in Figure 3.4 illustrates the geometric parameters associated with the DI. A higher DI value signifies superior cluster separation, indicating well-defined and distinct clusters.

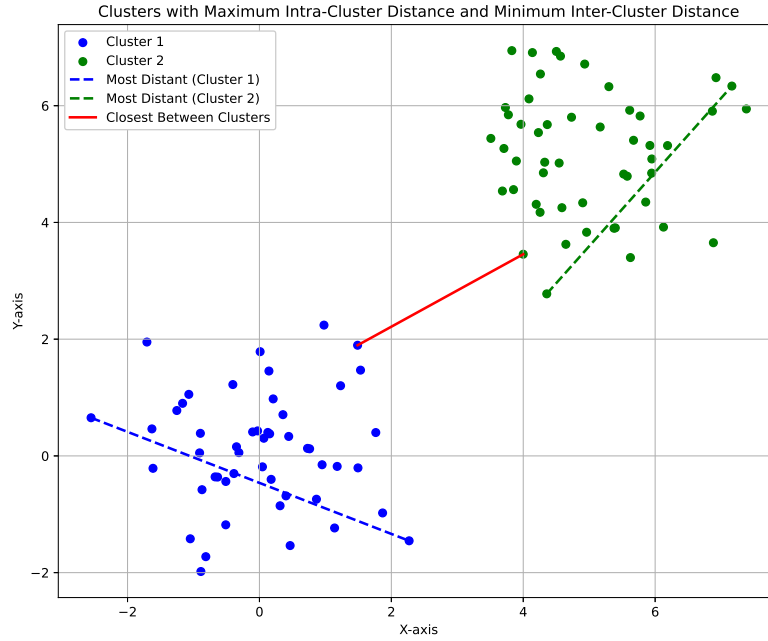


Fig. 3.4 Visualization of Dunn Index parameters in two-dimensional space, showing inter-cluster and intra-cluster distances.

Ideally, the DI should exceed unity, implying that the minimum distance separating clusters is substantially greater than the maximum diameter within any single cluster [136]. However, interpreting DI values in isolation can be challenging, as the index is influenced by factors such as the number of clusters, data distribution characteristics, and underlying dimensionality [137]. Therefore, contextual interpretation and comparative analysis are often required for meaningful evaluation.

The DI can be effectively integrated into the previously presented sequential monitoring framework to construct a control chart, as demonstrated in Figure 3.5. This chart facilitates real-time tracking of the clustering separability index over successive samples, where deviations beyond predefined control limits may indicate significant changes or anomalies in the underlying data stream process. By continuously monitoring DI values, practitioners can promptly detect shifts in process behavior, enabling timely interventions.

While the DI is effective in assessing cluster separation under favorable conditions, it may underestimate true separability in certain scenarios, particularly when data are distributed along elongated, thin structures or exhibit skewed cluster shapes. The performance of the DI improves substantially with balanced datasets, where clusters possess similar sizes, densities, and geometric configurations, thereby allowing the index to accurately capture both inter-cluster separation and intra-cluster compactness. In imbalanced scenarios, however, where cluster sizes or densities vary significantly, the index may yield distorted or misleading results, reducing the overall reliability of clustering assessments. Balanced datasets, therefore, enhance the interpretability and effectiveness of the DI, especially when employed to define rational subgroups within data streams for the purpose of constructing monitoring schemes and control charts. The UCL for the DI-based chart is typically estimated using appropriate simulation procedures that account for the statistical properties of in-control data.

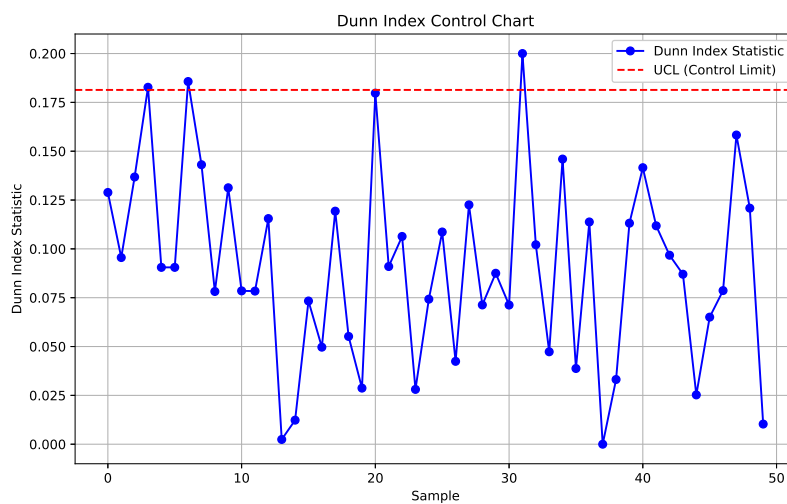


Fig. 3.5 Dunn Index control chart over time, with the red dashed line indicating the UCL.

Recent Family of Cluster Separability Indices

This subsection introduces a novel contribution of this thesis. To accurately evaluate how specific events or distributional shifts influence the behavior of an underlying data stream process, we propose and develop a new family of indices, collectively designated as *Skape* (SK). Unlike the classical Dunn Index presented above, which may exhibit limitations in scenarios with overlapping clusters or imbalanced data [85], **the SK family developed in this work** concentrates specifically on the detection of structural alterations and behavioral changes within streaming data environments. The *SK* indices are purposefully formulated to accentuate and quantify dissimilarities between samples acquired before and after a potentially disruptive occurrence. This is accomplished by exploiting straightforward yet statistically powerful characteristics extracted from the complete set of pairwise Euclidean distances computed between all observations in two sequential samples. Serving as a pragmatic and interpretable measure of separability, **our proposed SK indices** provide a comprehensive quantitative assessment of the degree of divergence between two data samples captured during different temporal intervals, which are conceptualized as distinct groups or clusters, denoted C_1 and C_2 . These clusters may or may not correspond to differing operational states, process regimes, or underlying system conditions. This analytical capability facilitates the establishment of a robust and adaptive control scheme designed to continuously monitor the separability between successive samples, thereby enabling practitioners to promptly identify potential shifts, anomalies, or emerging fault conditions in the data stream process. The comparative performance of the proposed SK indices against established metrics is rigorously evaluated in Chapter 4 through extensive simulation studies and real-world datasets.

In what follows, the formal mathematical definition of the *SK* indices is presented in detail, followed by a comprehensive discussion of their practical application within a RTM framework. These indices are constructed by synthesizing information derived from pairwise distance distributions, centroid separation metrics, and boundary extremes, all of which are essential for accurately quantifying the structural relationships and spatial configurations within sequential data. The indices are formally defined as:

$$SK_{max*} = \frac{(d_{cent*} + 1)^2}{d_{mean}} d_{max}$$

$$SK_{std*} = \frac{(d_{cent*} + 1)^2}{d_{mean}} d_{std}$$

where the constituent distance metrics are defined as follows:

- d_{mean} denotes the mean pairwise Euclidean distance calculated between all observations in two distinct samples, C_1 and C_2 , which contain n_1 and n_2 distinct data points, respectively. This average distance is determined using the following mathematical relation:

$$d_{mean} = \frac{1}{n_1 \cdot n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} d_{ij}.$$

This distance metric quantifies the average degree of separation between the two clusters under examination. A relatively low mean distance indicates either tight intra-cluster cohesion or a substantial degree of overlap between the clusters, suggesting that the samples may originate from similar underlying distributions. Conversely, a high mean distance suggests pronounced inter-cluster separation, which is a desirable characteristic for well-differentiated clustering configurations and indicates that the two samples likely represent distinct operational or behavioral states.

- d_{std} quantifies the variability or dispersion of all pairwise distances between observations in the two samples. It is calculated using the standard deviation formula:

$$d_{std} = \sqrt{\frac{1}{n_1 \cdot n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} (d_{ij} - d_{mean})^2}.$$

The variability of pairwise distances provides crucial insights into both cluster cohesion and the degree of separation between clusters. Clusters that exhibit tight internal cohesion combined with minimal separation from one another display small variability values, reflecting homogeneity in inter-point distances. In contrast, dispersed clusters or those that are well-separated yield larger variability values, reflecting greater heterogeneity in the spatial distribution of distances and indicating more distinct clustering structures.

- d_{max} represents the maximum pairwise distance observed between any point in cluster C_1 and any point in cluster C_2 , as visually depicted by the dashed line in Figure 3.6. This metric is formally defined as:

$$d_{max} = \max_{1 \leq i \leq n_1, 1 \leq j \leq n_2} d_{ij}.$$

This maximum distance metric reveals the spatial boundaries and extent of the clusters, and it is particularly sensitive to the presence of potential outliers

or extreme observations located at the peripheries of the data distributions. Well-separated clusters characterized by clear boundaries exhibit large d_{max} values, whereas overlapping or insufficiently separated clusters yield smaller values, indicating that the furthest points in each cluster are relatively close to one another.

- d_{cent*} represents the Euclidean distance between the centroids (mean vectors) of the two clusters, as illustrated by the solid black line in Figure 3.6. This centroid distance serves as a direct measure of the overall positional separation between the two samples in the feature space. Well-separated clusters with distinct central tendencies exhibit large d_{cent*} values, whereas clusters that overlap significantly or are inadequately separated are characterized by smaller centroid distances, reflecting spatial proximity in their average locations.

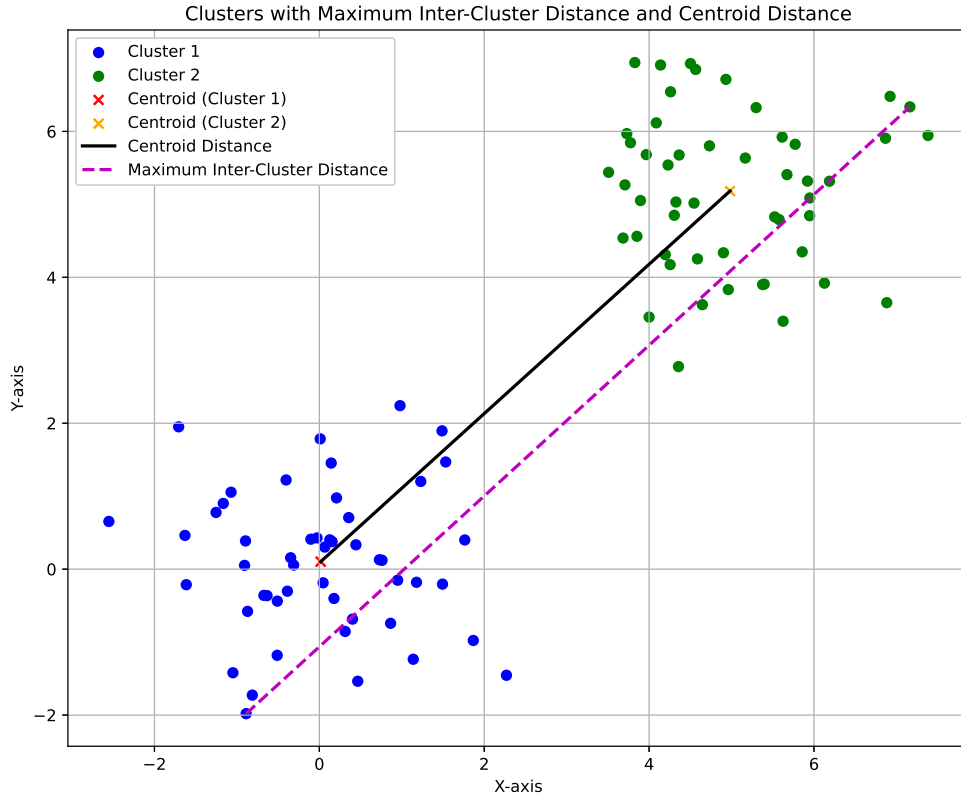


Fig. 3.6 Illustration of the fundamental geometric quantities associated with the *Skape* family of indices. The solid black line indicates the distance between the two cluster centroids, while the dashed line represents the maximum pairwise distance between the two most distant points across the clusters.

It is important to note that Figure 3.6 provides a visual illustration of the basic geometric quantities and spatial relationships associated with the recently introduced family of indices, thereby facilitating intuitive understanding of their construction. Meanwhile, Figure 3.7 presents a control chart utilizing the SK_{max}^* index over successive time intervals. In this chart, the red dashed line indicates the UCL, and values exceeding this threshold suggest potential process instability or the occurrence of a structural shift, warranting further investigation and possible corrective action. Both indices fundamentally rely on a functional relationship involving the ratio d_{cent^*}/d_{mean} , which is designed to detect whether the two samples, when interpreted as clusters, exhibit a meaningful structural difference. Specifically, this ratio helps determine whether the clusters are compact and well-separated or whether they overlap signif-

icantly. When clusters are well-separated and distinct, the ratio approaches unity, indicating that the centroid distance is comparable to the mean pairwise distance. Conversely, when clusters overlap substantially, the ratio approaches zero, reflecting the fact that the centroids are much closer than the average distance between individual observations.

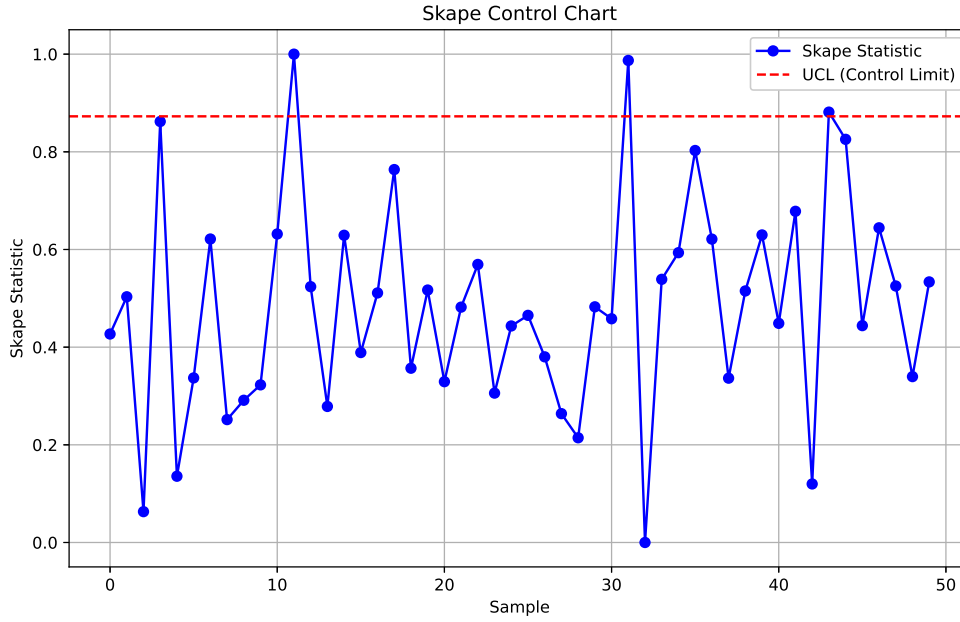


Fig. 3.7 Control chart based on the SK_{max}^* index over sequential periods. The red dashed line marks the UCL, which signifies possible process instability if exceeded, thereby prompting further investigation and the implementation of corrective measures.

The addition of 1 to the term d_{cent}^* in the numerator of the SK family definitions serves a critical role in ensuring numerical stability and preventing the metric from becoming overly sensitive or collapsing to near-zero values when d_{cent}^* is small. Without this adjustment, and given the squared factor applied to this term, the metrics could approach values close to zero when the centroids of the clusters are very close to each other, regardless of the values of variability d_{std} or boundary extremes d_{max} . This would result in significantly under-representing potentially important overall structural differences between clusters, particularly in situations where substantial internal spread or significant boundary separation is present. By adding 1, the metrics maintain a baseline level of sensitivity to the underlying cluster structure, even in cases where centroid separation is minimal or negligible, thereby ensuring robustness across diverse data configurations.

Moreover, squaring the term $(d_{cent*} + 1)$ serves to amplify its contribution to the overall metric, ensuring that increases in centroid separation exert a disproportionately greater impact on the final index value. This design choice deliberately highlights the central role of centroid separation in distinguishing between clusters and detecting structural changes. Simultaneously, the squaring operation mitigates the risk of overemphasizing the contributions of d_{max} and d_{std} , which reflect boundary extremes and internal dispersion, respectively, particularly in cases where d_{cent*} is relatively small. This balanced formulation ensures that the indices remain responsive to multiple complementary aspects of cluster geometry, thereby providing a more holistic and nuanced assessment of cluster quality and separability.

Together, the two metrics provide a robust, flexible, and comprehensive framework for assessing the structural relationships between two samples treated as distinct clusters. The SK_{max*} index excels in capturing boundary extremes and detecting clusters with extended spatial ranges or the presence of outliers, making it particularly sensitive to abrupt distributional shifts or the emergence of extreme observations. In contrast, the SK_{std*} index highlights overall variability and internal dispersion patterns, offering enhanced flexibility to address both abrupt shifts and gradual, subtle changes in clustering behavior. As a result, the inclusion of both versions within the monitoring framework ensures that the system remains sensitive to a diverse array of scenarios and operational conditions. This dual-index approach allows practitioners to detect, analyze, and respond effectively to various types of structural shifts, anomalies, and behavioral changes in dynamic data streams with greater precision, reliability, and interpretability.

Modifications of Separability Indices Given that the current statistical approach examines consecutive samples in a sequential manner without explicitly incorporating a reference sample that may be available from Phase I monitoring, an alternative and enhanced version of the previously mentioned indices is also presented in this paragraph. This revised approach introduces a pooled reference cluster constructed from a stable baseline sample obtained during Phase I, thereby anchoring the monitoring process to an established in-control distribution. By incorporating this reference information, the modified indices enable more sensitive detection of deviations from historically stable process behavior. The modified indices are formally defined as:

$$SK_{max} = \frac{(d_{cent} + 1)^2}{d_{mean}} d_{max}$$

$$SK_{std} = \frac{(d_{cent} + 1)^2}{d_{mean}} d_{std}$$

where d_{cent} now represents the Euclidean distance between the pooled mean of the reference cluster (constructed from Phase I in-control data) and the mean of the centroids of the two consecutive samples currently under evaluation (treated as discrete clusters). This modification enables the indices to effectively leverage historical in-control information, thereby enhancing their sensitivity to deviations from established baseline behavior and improving the overall robustness, reliability, and interpretability of the monitoring scheme in real-time and high-frequency data stream applications.

3.5 Simulation and Performance Analysis

This section presents a comprehensive suite of simulation studies to evaluate the robustness, sensitivity and adaptability of the proposed monitoring methodologies within high-dimensional data stream environments. Building upon the analytical challenges outlined in preceding sections, including the curse of dimensionality, evolving distributional patterns in streaming data, and the stringent computational constraints inherent to RTM applications, the following simulation experiments are specifically designed to rigorously test the performance of the proposed SK family of indices alongside well-established and widely adopted metrics, such as Hotelling's T^2 statistic and the Dunn Index. By methodically examining a diverse range of operational scenarios, including mean shifts of varying magnitudes under different degrees of inter-variable correlation, and by considering both MND and non-Gaussian random distributions, the effectiveness of these metrics in detecting process anomalies and ensuring reliable, continuous monitoring across different statistical frameworks and process configurations is thoroughly evaluated.

The simulation study is structured as follows. Initially, the simulation setup is presented in detail, including a comprehensive description of the underlying distributional frameworks employed, the specific monitoring scenarios evaluated, and the two distinct monitoring approaches implemented for comparative analysis. Subsequently, an in-depth analysis of the simulation results is presented and divided into separate subsections addressing MND contexts and random distribution contexts. This organizational structure facilitates a clear, methodical comparison of the metrics' performance characteristics, detection capabilities, and computational efficiency across varying data generation mechanisms and operational conditions, thereby providing actionable

insights into the practical applicability and relative advantages of each monitoring approach in modern data-intensive environments.

3.5.1 Simulation Setup

The simulation setup is designed to generate diverse operational scenarios that effectively capture complex inter-variable relationships, structural variability, and evolving process dynamics, thereby enabling a more realistic and comprehensive representation of real-world data patterns commonly encountered in industrial and data-intensive monitoring applications. By incorporating different types of structural modifications, varying degrees of inter-variable correlation, and controlled distributional perturbations, the present study rigorously evaluates how variations in fundamental data characteristics impact the analytical performance, detection sensitivity, and reliability of the competing monitoring methodologies under investigation.

The simulation process encompasses a series of carefully controlled mean shifts and variance alterations introduced at predetermined points, designed to assess the effectiveness and responsiveness of anomaly detection methods across a spectrum of fault magnitudes and process disturbances. These controlled perturbations ensure that even subtle, gradual changes in process behavior can be identified and distinguished from normal operational variability, thereby testing the robustness of each monitoring approach under realistic operational conditions. Particular attention is given to scenarios involving high-dimensional feature spaces, where traditional monitoring techniques often encounter difficulties due to sparsity, computational complexity, and the deterioration of statistical assumptions.

Furthermore, two fundamentally distinct monitoring strategies are employed throughout the simulation framework to facilitate comprehensive comparative analysis. The first strategy adopts a reference-based monitoring approach, in which newly acquired observations are sequentially compared against a stable historical reference distribution established during Phase I using in-control data. This approach leverages baseline information to detect deviations from established norms and is particularly effective when a reliable reference dataset is available. The second strategy implements a sequential monitoring approach, which continuously evaluates changes and structural differences between consecutive data samples or time windows, without explicitly relying on a fixed historical reference. This adaptive strategy is particularly suited for dynamic environments where process characteristics may evolve over time, and where maintaining a static reference may not be practical or representative of current operational states. Together, these two strategies provide complementary perspectives

on process stability and enable a thorough evaluation of monitoring performance across diverse data stream configurations and operational contexts.

Distributional Framework

Multivariate Normal Distribution The probability density function (pdf) of a MND characterized by a mean vector $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{\Sigma}$, evaluated at a given point $\mathbf{x}$ in d -dimensional space, is formally expressed as:

$$f(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{d}{2}} \det(\boldsymbol{\Sigma})^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right),$$

where $\boldsymbol{\mu}$ denotes the mean vector encapsulating the central tendency of each variable, $\boldsymbol{\Sigma}$ represents the covariance matrix describing the variance structure and inter-variable dependencies, d specifies the dimensionality of the data space, $\det(\boldsymbol{\Sigma})$ is the determinant of the covariance matrix, and $\exp$ denotes the exponential function.

This mathematical expression quantifies the likelihood of observing a particular data point $\mathbf{x}$ given the specified mean vector and covariance structure of the MND, thereby forming the foundation for parametric statistical inference and hypothesis testing in multivariate contexts. In practical computational implementation, leveraging specialized numerical libraries such as NumPy in the Python programming environment confers significant advantages in terms of efficiency, numerical precision, and reproducibility when generating random samples from the MND [138]. The highly regarded and widely utilized NumPy random sampling function serves as an essential computational tool, efficiently generating high-quality multivariate normal samples while seamlessly accepting the mean vector and covariance matrix as input parameters. Once the distributional parameters have been appropriately defined and validated, and the requisite computational techniques have been applied, the resulting samples become readily available for detailed statistical analysis, exploratory visualization, and hypothesis testing. These samples provide valuable empirical insights into the underlying distributional structure, offering representative snapshots that facilitate the exploration of complex inter-variable relationships, correlation structures, and dependency patterns. This empirical analysis serves as a robust foundation for informed decision-making processes, enabling researchers and practitioners to derive actionable insights and statistical inferences from the resulting data.

Hadamard Product Distributions In real-world operational and data-driven scenarios, empirical datasets rarely conform strictly to a single, well-defined parametric

distribution. To address this complexity, Hadamard product distributions provide a powerful and flexible mathematical framework for modeling such multifaceted data structures and capturing their inherent complexities. One effective method for constructing a complex random distribution is by computing the element-wise (Hadamard) product of two distinct multivariate distributions, such as combining a MND with another random multivariate distribution possessing different statistical properties. This multiplicative process facilitates the introduction of specific controlled perturbations, stochastic noise components, or structural modifications into the reference distribution, thereby creating more realistic and representative models of empirical data that account for both continuous stochastic signals and discrete event-driven phenomena. Two particularly notable and well-studied examples of Hadamard product multivariate distributions are the MRD [139], which is frequently employed in signal processing and communications applications, and the Multivariate Poisson Distribution (MPD) [140], which is commonly used in count data modeling and event occurrence analysis. By strategically leveraging such product distributions, researchers can construct more sophisticated and contextually appropriate models that effectively capture the intricate structural complexities, non-Gaussian behaviors, and interaction effects inherent in real-world datasets, where multiple underlying stochastic processes interact in ways that cannot be fully or adequately described by reliance on a single parametric distribution alone.

Product of Samples:

The matrix $\mathbf{X}_{\text{MRND}}$ represents the Hadamard (element-wise) product of the matrix $\mathbf{X}_{\text{MND}}$, containing samples drawn from a MND, and a random distribution matrix $\mathbf{X}_{\text{RND}}$, containing samples from an alternative distribution. This operation is formally expressed as:

$$\mathbf{X}_{\text{MRND}} = \mathbf{X}_{\text{MND}} \odot \mathbf{X}_{\text{RND}},$$

where $\odot$ denotes the element-wise multiplication operator applied entry-wise across corresponding matrix elements. Two distinct cases are considered based on the specific choice of the random distribution employed. In the first case, when $\mathbf{X}_{\text{RND}}$ follows a MRD with scale parameter $\sigma = 1$, the resulting product matrix is denoted as:

$$\mathbf{X}_{\text{MRND1}} = \mathbf{X}_{\text{MND}} \odot \mathbf{X}_{\text{MRD}},$$

where $\mathbf{X}_{\text{MRD}}$ represents the matrix containing Multivariate Rayleigh distributed samples. In the second case, when $\mathbf{X}_{\text{RND}}$ follows a MPD with rate parameter $\lambda = 2$, the resulting product matrix is denoted as:

$$\mathbf{X}_{\text{MRND2}} = \mathbf{X}_{\text{MND}} \odot \mathbf{X}_{\text{MPD}},$$

where $\mathbf{X}_{\text{MPD}}$ represents the matrix containing MPD samples. In both cases, the Hadamard product operation is applied between the matrix $\mathbf{X}_{\text{MND}}$ and the corresponding random distribution samples, resulting in hybrid distributions that combine the properties of their constituent components.

The following subsections present comprehensive results from simulation studies conducted to rigorously demonstrate the efficacy, sensitivity, and robustness of the statistical metrics devised to evaluate and monitor samples drawn from these complex random distributions, including those constructed through the multiplicative combination of different multivariate distributions. These simulation results elucidate the relative strengths, limitations, and operational characteristics of the metrics employed, showcasing their applicability, performance, and interpretability in practical monitoring contexts. The detailed comparative analysis of the simulation outcomes provides a thorough and nuanced understanding of how these metrics, including those specifically designed for product distributions, can substantially enhance analytical capabilities and improve anomaly detection performance when applied to multifaceted, non-Gaussian, and structurally complex datasets.

Hadamard Product of Multivariate Normal and Rayleigh Distributions

The relationships between variables are examined and quantified through their correlation structure, which manifests differently across distinct types of multivariate joint probability distributions. A joint distribution formally describes the probabilistic behavior and simultaneous occurrence patterns of two or more random variables. A fundamental aspect of such distributions concerns how the variables relate to one another, and correlation serves as a widely used measure of their linear dependence or association. Specifically, this analysis compares the well-established MND, a natural generalization of the univariate normal distribution to multiple dimensions, with a more structurally complex distribution constructed by combining the MND with the MRD through the Hadamard product operation. The bivariate MND, representing a special case of the general MND restricted to two variables, is characterized by its symmetric, bell-shaped density surface, and its geometric and statistical behavior under varying degrees of correlation is well understood and extensively documented in the statistical

literature. When visualizing only two dimensions of a higher-dimensional MND through marginal or conditional projections, as the correlation coefficient increases between those two selected dimensions, the distribution's probability density contours undergo a predictable geometric transformation from an approximately circular shape (indicating independence or low correlation) to an increasingly elongated elliptical shape (reflecting strong linear association). This visual transformation directly represents the growing statistical tendency of those two variables to exhibit closely related or dependent behavior. The second distribution under consideration is formed by computing the element-wise Hadamard product of independent samples from a MND and a MRD. The MRD, in contrast to the symmetric normal distribution, is characterized by its asymmetric, non-negative support, and is frequently employed to model magnitude-related quantities, signal amplitudes, or radial distances in various scientific and engineering applications. Combining the MRD with a MND in this multiplicative manner creates a joint distribution whose statistical properties, geometric characteristics, and correlation structure are not as immediately apparent or analytically tractable as those of the MND alone. The presence of the Rayleigh multiplicative component is expected to significantly alter the resulting distribution's overall behavior (see Figure 3.8), especially in terms of how correlation between any two dimensions influences the overall geometric shape and density concentration when visualized through two-dimensional projections or scatter plots.

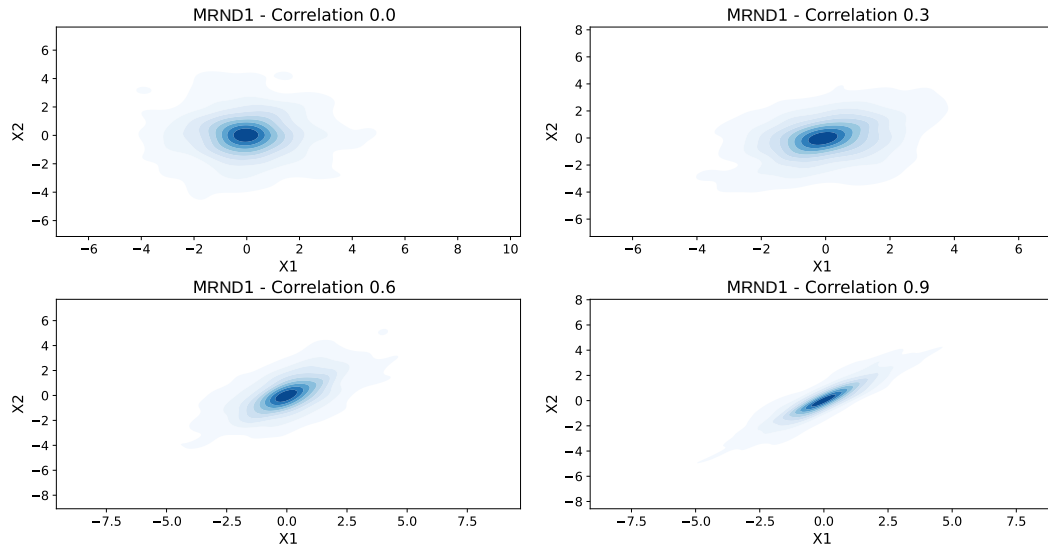


Fig. 3.8 Hadamard product of Multivariate Normal and Rayleigh Distributions under various degrees of correlation, illustrating the structural transformation as inter-variable correlation increases.

Hadamard Product of Multivariate Normal and Poisson Distributions Extending the previous comparative analysis of joint distributions involving the MND and MRD, this paragraph examines an analogous scenario in which the well-established MND is compared with a more structurally complex distribution derived from combining the MND with the MPD through the Hadamard product operation. The MPD is a discrete, non-negative distribution commonly employed to model count data, event occurrences, and integer-valued random phenomena, and it introduces pronounced asymmetry and discretization effects when combined multiplicatively with a continuous normal distribution. This Hadamard product operation results in a unique hybrid multivariate distribution whose statistical properties, geometric characteristics, and correlation behavior differ substantially from those of the standard MND. It is anticipated that the inherently discrete and non-negative nature of the Poisson multiplicative component will substantially influence the resulting distribution's shape, support structure, and its response to varying degrees of correlation between the normal components. The visualization presented in Figure 3.9 illustrates these differences, comparing the impact of increasing correlation between two selected dimensions on the standard MND with its corresponding effect on this combined normal-Poisson product distribution. This graphical comparison allows for direct observation of how the distinctive structural characteristics of the Poisson distribution interact with the correlation present in the normal components, leading to density contours and scatter patterns that deviate considerably from the typical smooth elliptical forms observed in bivariate MND projections.

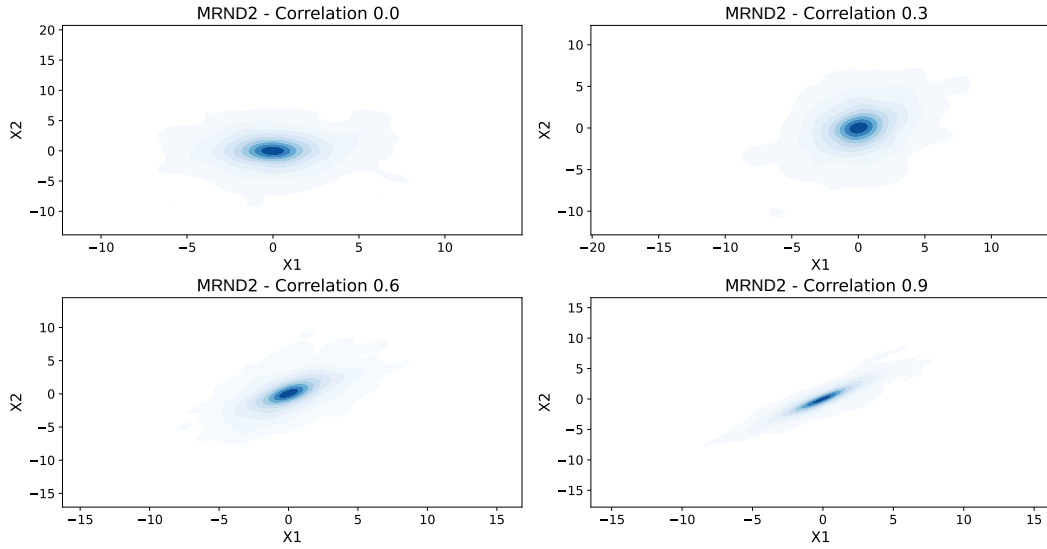


Fig. 3.9 Hadamard product of Multivariate Normal and Poisson Distributions under various degrees of correlation, highlighting the discretization effects and structural deviations from standard multivariate normality.

The geometric characteristics and topological features of contour ellipses and density plots (as illustrated in Figure 3.9) provide essential qualitative and quantitative insights into the nature of the underlying probability distribution and its support structure. Continuous probability distributions are typically associated with smooth, differentiable, and continuous contour lines, reflecting gradual and uninterrupted variations in probability density across the domain. In stark contrast, discrete distributions often exhibit angular, segmented, or irregular contour patterns, sometimes described as “boxy” or grid-like, resulting from probability mass being concentrated at distinct, isolated points rather than being smoothly distributed across a continuum. This discretization leads to abrupt transitions in density representation and produces contour structures characterized by sharp edges, right-angled boundaries, or step-like formations. This phenomenon arises fundamentally due to the absence of smooth interpolation between discrete values in such distributions, producing contour geometries with pronounced discontinuities or lattice-like formations. Therefore, the presence of such angularity, segmentation, and grid-like structure in contour plots serves as a key visual and diagnostic indicator of underlying discretization in the data, enabling practitioners to infer the nature of the data-generating process directly from graphical representations.

Overview of Key Parameters

The simulation process is implemented using the Python programming environment, with a comprehensive set of global parameters explicitly defined to ensure reproducibility, consistency, and controlled experimental conditions throughout the entire simulation procedure. The set of correlation coefficients representing the degree of linear association among all pairs of variables, denoted as corr (also represented as ρ), is specified as $\rho \in \{0.0, 0.3, 0.6, 0.9\}$, thereby representing the varying levels of correlation strength considered in the comparative analysis. These values span the spectrum from complete independence ($\rho = 0.0$) to strong positive correlation ($\rho = 0.9$), enabling systematic evaluation of metric performance under diverse dependence structures.

The covariance matrix Σ is constructed by setting the diagonal elements to unity, representing standardized unit variance for each variable, thereby ensuring comparability across dimensions. The off-diagonal elements are populated uniformly with the correlation coefficients ρ , indicating the strength and direction of the linear relationships between the variables. This symmetric structure is formally expressed as:

$$\Sigma = \begin{bmatrix} 1 & \rho & \rho & \dots & \rho \\ \rho & 1 & \rho & \dots & \rho \\ \rho & \rho & 1 & \dots & \rho \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho & \rho & \rho & \dots & 1 \end{bmatrix}$$

This equicorrelated covariance structure, also known as compound symmetry, simplifies the analysis while maintaining realistic inter-variable dependency patterns commonly encountered in industrial process monitoring and multivariate data stream applications.

The dimensionality of the data are varied according to the Fibonacci sequence, with dim (also denoted as d) taking values from the set $d \in \{2, 3, 5, 8, 13, 21, 34, 55, 89, 144\}$, representing the number of dimensions or variables in the multivariate datasets under investigation. This choice of dimensionality progression is particularly advantageous, as the Fibonacci sequence provides a natural, exponentially growing sequence that effectively spans from low-dimensional to extremely high-dimensional scenarios, thereby enabling comprehensive evaluation of how monitoring performance scales with increasing dimensionality and facilitating the assessment of the curse of dimensionality effects on detection sensitivity.

For the mean shift factor employed to simulate out-of-control conditions, the magnitude of the shift μ_{shift} is defined as the 75th percentile of the in-control distribution's

standard deviation along each dimension, applied to the mean vector, which is initially set to the zero vector $\mu = [0, 0, \dots, 0]$. This shift magnitude ensures a moderate yet detectable deviation from the in-control state, providing a realistic scenario for evaluating anomaly detection capabilities. The number of observations per cluster in the data stream simulation is consistently set to $\text{rows} = 10$, reflecting the rational subgrouping strategy discussed previously and ensuring computational efficiency while maintaining sufficient statistical information for separability assessment. The total number of independent simulation replications is set to $\text{sim} = 100,000$, thereby ensuring robust statistical estimation of performance metrics, control limits, and detection probabilities with minimal Monte Carlo error.

For scenarios involving the generation of random samples from a MRD, the scaling parameter is consistently defined as $\sigma = 1$, ensuring standardized variability and facilitating comparability across distributional frameworks. Similarly, for the MPD, the rate parameter governing the expected count per dimension is set to $\lambda = 2$, providing a moderate level of discrete variability suitable for realistic count-based process modeling. Together, these carefully selected parameter values establish a comprehensive and well-controlled experimental design that enables rigorous evaluation of the recently proposed monitoring methodologies across a wide range of operational conditions, distributional assumptions, and dimensionality levels.

Scenarios Evaluated

Due to the considerable computational demands and the extensive time required to comprehensively examine all conceivable operational scenarios and distributional configurations, this study focuses specifically on the most prevalent and practically significant types of process abnormalities. These include mean shift anomalies and the capacity to identify irregularities in combined or hybrid distributed data streams formed through Hadamard product operations. These scenarios are representative of common fault conditions encountered in industrial monitoring applications and provide a rigorous basis for evaluating the detection capabilities of the proposed methodologies.

For testing the performance of the detection algorithms during Phase II monitoring, a pair of sequential samples in the data stream is generated from distributions sharing identical shape parameters and spatial orientation but located at slightly different positions in the multivariate feature space. This controlled mean shift is designed to mimic potential anomalies or process disturbances that may arise in real-world operational contexts. The data generation process commences with the creation of an initial dataset drawn from a MND characterized by a predefined mean vector

and covariance structure. Each data point $\mathbf{x}_i$ in the dataset, with $i = 1, 2, \dots, N$, is independently sampled from this distribution within a d -dimensional space, yielding a comprehensive dataset $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ representative of in-control process behavior.

Subsequently, the 75th percentile (also referred to as the third quartile, Q3) is computed along each dimension of the dataset. This computation involves determining the value below which 75% of the observed data points fall along each respective dimension, resulting in a percentile vector $\mathbf{x}_{75} = (x_{75}^{(1)}, x_{75}^{(2)}, \dots, x_{75}^{(d)})$, which characterizes the upper portion of the in-control distribution along each variable. The mean vector $\boldsymbol{\mu}$ of the original dataset is then intentionally shifted to align with this 75th percentile point $\mathbf{x}_{75}$ along each dimension, thereby simulating a moderate yet detectable process shift. This shift is formally denoted by a shift vector $\mathbf{s}$, calculated as the difference between the percentile vector and the original mean vector, expressed mathematically as $\mathbf{s} = \mathbf{x}_{75} - \boldsymbol{\mu}$.

The shifted dataset $\mathcal{D}_{\text{shifted}}$ is subsequently generated by redefining the mean parameter of the MND to $\boldsymbol{\mu}_{\text{new}} = \boldsymbol{\mu} + \mathbf{s} = \mathbf{x}_{75}$, while maintaining the original covariance structure Σ unchanged. This results in a new dataset where the mean vector is aligned with the 75th percentile point along each dimension, effectively simulating an out-of-control condition (see Figure 3.10 for a visual representation).

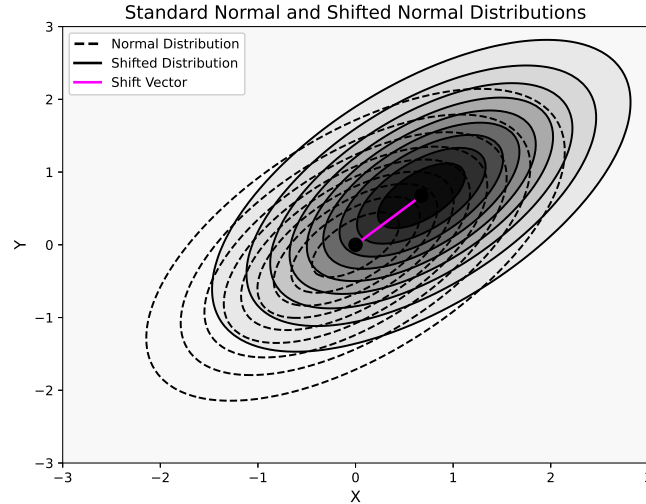


Fig. 3.10 Illustration of a mean shift from the initial sample (represented by dashed contour ellipse) to the shifted sample (represented by solid contour ellipse). The length of the shift vector (depicted as a magenta line) illustrates the magnitude of the shift $\mathbf{s}$ toward the 75th percentile location in the feature space.

In the case of hybrid random distributions constructed through Hadamard product operations (i.e., MND combined with MRD or MPD), the shift in the mean is applied exclusively to the MND component, while the parameters of the multiplicative distribution (Rayleigh or Poisson) remain unchanged. This design choice isolates the effect of the mean shift within the continuous normal component, allowing for clearer interpretation of detection performance under hybrid distributional conditions.

The primary objective of this controlled shift procedure is to assess the ability of the monitoring schemes to detect changes in the underlying distribution by deliberately shifting the mean vector to align with a specific upper percentile along each dimension. This intentional shift is expected to amplify in magnitude as the dimensionality of the dataset increases, due to the geometric properties of high-dimensional spaces, leading to increasingly significant separation between the original in-control and shifted out-of-control distributions. This enhanced separation improves the discriminative power of statistical tests and monitoring algorithms designed to detect distributional changes, making even subtle or gradual shifts more apparent and more readily detectable. By sequentially analyzing sample pairs under these controlled mean shift conditions, the effectiveness of the proposed metrics in detecting simulated anomalies is rigorously evaluated by comparing the computed metric values to the previously determined control limits and percentile thresholds established during Phase I monitoring. This approach provides a comprehensive and quantitative basis for assessing detection sensitivity, false alarm rates, and overall monitoring performance across diverse operational scenarios and dimensionality levels.

Monitoring Approaches

This subsection presents two fundamentally distinct yet complementary approaches for SPM, each designed to assess the control status of incoming samples using different analytical frameworks, reference structures, and underlying statistical assumptions. These approaches enable comprehensive evaluation of monitoring performance under diverse operational conditions and facilitate rigorous comparative assessment of the proposed metrics.

Approach 1 (“RP1S”): Exploiting a Reference Phase I Sample This approach utilizes a pooled mean and pooled covariance matrix derived from a comprehensive aggregation of archived in-control samples collected during Phase I. This pooled mean is calculated by consolidating data from a cohort of historical samples to obtain a statistically representative average that characterizes the central tendency of the in-

control process state. The pooled mean vector, denoted as $\boldsymbol{\mu}_{\text{pooled}}$, serves as a stable and reliable benchmark against which new observations are compared. In addition to the pooled mean, a pooled covariance matrix Σ_{pooled} is computed, capturing the inherent variability, dispersion patterns, and inter-variable dependencies within the historical in-control dataset. This covariance matrix encapsulates the correlation structure and variance components that characterize normal process behavior under stable operating conditions. During the Phase II monitoring phase, each sample under observation is compared to both the pooled mean and the pooled covariance matrix to assess deviation from established baseline behavior. The mean of the current sample, also referred to as the sample centroid, is evaluated to determine its control status by quantifying its statistical distance from the pooled reference mean. This methodology inherently assumes that the pooled mean, in conjunction with the pooled covariance matrix, serves as a stable, representative, and reliable benchmark for assessing the control status of new samples entering the monitoring system. The stability and reliability of this reference benchmark are critical to ensuring accurate anomaly detection and minimizing false alarm rates. In the context of the RP1S approach, the SK indices are formulated based on the distance between the pooled mean and the average of the centroids of the sample pairs currently under evaluation. This design ensures a methodologically valid and statistically consistent comparison with the T^2 metric, as both statistics utilize the same reference mean and covariance structure for deviation assessment, as also noted in paragraph 3.4.2. This alignment enhances interpretability and ensures that performance comparisons between metrics are conducted on an equitable basis.

Approach 2 (“CONS”): Monitoring Consecutive Samples Comparison In contrast to the reference-based RP1S approach, the CONS approach focuses specifically on the direct comparison of sequentially paired samples for RTM purposes. This approach emphasizes the temporal relationships, structural differences, and evolving patterns between consecutive samples rather than comparing individual sample statistics to a fixed, static reference established during Phase I. This fundamental distinction differentiates the CONS approach from the RP1S approach, where both the pooled mean and the pooled covariance matrix are integral components of the monitoring framework. The CONS approach is particularly well-suited for dynamic or non-stationary environments where process characteristics may gradually evolve over time, and where maintaining a fixed reference may not fully capture the current operational state.

Nevertheless, both approaches are designed to provide a robust and rigorous comparison against a reliable and well-established monitoring statistic, specifically the Hotelling T^2 statistic. To ensure consistency and comparability across both methodologies, the pooled covariance matrix derived from the RP1S approach is also utilized within the CONS framework. This pooled covariance matrix remains constant between the two approaches, thereby ensuring that differences in monitoring performance are attributable to the choice of reference structure (pooled mean versus consecutive comparison) rather than differences in variance estimation. Given its essential role in the computation of the T^2 statistic and the calculation of Mahalanobis-type distances, maintaining the stability, accuracy, and validity of the covariance matrix across both monitoring frameworks is crucial. This consistency helps prevent potential statistical issues, such as a singular (non-invertible) covariance matrix or unreliable estimations of inter-variable relationships, which could lead to biased results, inflated false alarm rates, or reduced detection sensitivity.

For notational convenience and clarity throughout the remainder of this study, the symbol “*” is employed to indicate which monitoring approach has been applied to the respective statistics. Specifically, the asterisk signifies that the simulation follows the logic of the second approach, CONS, which focuses on consecutive sample comparisons. The DI, by its inherent definition as a cluster separability metric, is applicable only within the consecutive comparison framework and is therefore not included in the RP1S approach. Thus, for the first approach (RP1S), the statistical measures employed are: RP1S approach = $\{T^2, SK_{max}, SK_{std}\}$, whereas for the second approach (CONS), the statistical measures are: CONS approach = $\{T_*^2, DI, SK_{max*}, SK_{std*}\}$. This clear notational distinction facilitates transparent interpretation of simulation results and ensures that comparisons between approaches are conducted rigorously and unambiguously.

3.6 Results

To comprehensively understand the behavior and statistical characteristics of the monitoring metrics under varying distributional frameworks, correlation structures, and dimensionality levels, four key statistical summary measures are reported throughout the analysis: the mean, standard deviation, minimum, and maximum values. These descriptive measures collectively provide a comprehensive statistical characterization of the empirical distribution of each metric, capturing central tendency, dispersion, range, and variability across the simulation replications. The mean quantifies the

expected value or typical magnitude of the metric under the specified conditions, while the standard deviation reflects the degree of variability or uncertainty in metric values across repeated experiments. The minimum and maximum values define the observed range and reveal the presence of extreme values or outliers that may influence monitoring performance.

To support the interpretation and facilitate deeper understanding of these descriptive statistics, two complementary trend assessment methodologies were implemented and applied across all scenarios. The first method is the rate of change analysis, which captures local, short-term variations by calculating the differences between consecutive values of the metric across increasing dimensionality or correlation levels. This approach is particularly useful for identifying abrupt transitions, threshold effects, or non-monotonic behavior patterns that may not be immediately apparent from global trend analysis. The second method is linear regression analysis, which provides a global, long-term perspective by fitting a best-fit linear model to the data and interpreting the slope coefficient as an indicator of the overall trend direction and magnitude. A positive slope suggests increasing metric values with dimensionality or correlation, while a negative slope indicates a decreasing trend. The combination of these two analytical approaches enables a nuanced understanding of both local fluctuations and global tendencies in metric behavior.

The trends derived from these complementary methods are based on the values of the statistics presented in the detailed tables included in the appendix, which further illustrate the fluctuations, distributional characteristics, and overall behavior of the metrics across the full range of simulation parameters. These tabulated results provide a granular, quantitative foundation for the graphical and analytical interpretations presented in the main text, ensuring transparency, reproducibility, and comprehensive documentation of the simulation outcomes.

The subsequent subsections present the comprehensive results of simulations conducted for both Phase I and Phase II monitoring contexts. Phase I findings, detailed extensively in the appendix tables, offer a granular, high-resolution view of the data, including control limit estimation, distributional properties under in-control conditions, and the establishment of baseline performance benchmarks. Phase II results, in contrast, are presented primarily in graphical form, focusing on the detection performance, sensitivity, and false alarm characteristics of individual metrics under controlled out-of-control scenarios. This graphical presentation allows for a more intuitive, accessible, and comparative understanding of metric behavior across different operational scenarios,

distributional frameworks, and monitoring approaches, thereby facilitating actionable insights into the relative advantages and limitations of each proposed methodology.

The simulation results for both the in-control case (Phase I) and the monitoring phase (Phase II) are described and discussed with respect to multivariate normally distributed data. Phase I establishes the baseline performance characteristics of various statistical metrics, including the mean, standard deviation, and various percentiles, under the assumption of stable process conditions with no deviations from expected statistical behavior. Phase II examines how these metrics respond when a controlled shift in the mean of the distribution is introduced, thereby simulating an out-of-control condition. The primary objective is to rigorously assess the effectiveness, sensitivity, and robustness of these metrics in detecting such changes, providing comprehensive insights into their practical utility and relative performance in the context of MND monitoring.

3.6.1 Phase I: In-Control Conditions

Multivariate Normal Distribution This subsection characterizes the in-control behavior of Hotelling's T^2 , the DI, and the SK family of indices across correlation levels ρ and dimensionalities d , grounding each claim in the Appendix summary tables (Table 3.2). Consistent with degrees-of-freedom effects, the mean of T^2 increases strongly with d , whereas DI exhibits a milder, sometimes plateauing increase, and RP1S SK shows non-monotonic patterns at low d or high ρ ; in contrast, the Sequential (CONS) SK increases robustly with d . As ρ increases, T^2 and SK generally rise, while DI declines due to increased compactness and overlap, with occasional non-monotonic responses for T^2 and RP1S SK . These metric tendencies align with geometric distance effects: mean and minimum pairwise distances decrease, maxima initially increase and then stabilize, and the standard deviation broadens as ρ grows, as visualized in Figures 3.11 and 3.12 and summarized in Table 3.1.

Observed Effects of Increasing Correlation on Pairwise Distances				
Distance Type	Mean	Min	Max	Std
Intra-cluster	↓ Decreases	↓ Decreases	↑ Increases	↑ Increases
Inter-cluster	↓ Decreases	↓ Decreases	↑ Increases	↑ Increases

Table 3.1 Directional trends in pairwise distance statistics with increasing ρ , supporting the metric responses discussed in the text.

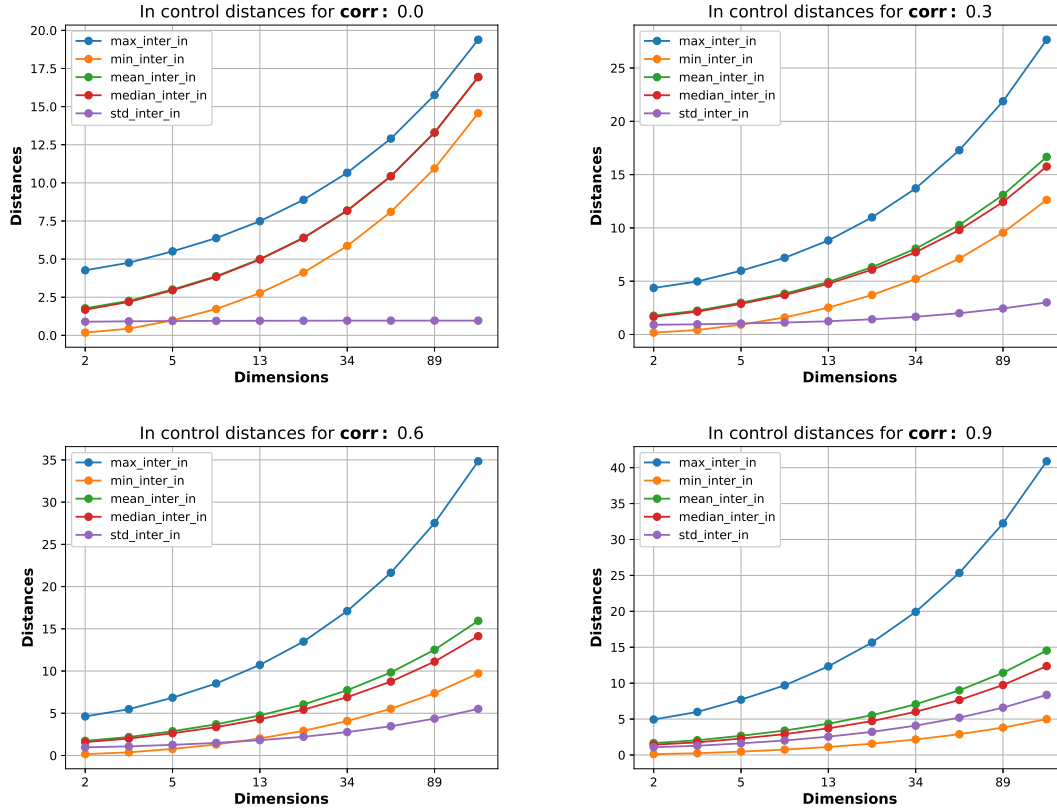


Fig. 3.11 Inter-cluster pairwise distances in Multivariate Normal Distribution across d and ρ , illustrating contraction of means/minima and broadening of extremes with correlation.

Hadamard Product: Rayleigh - Normal The $\mathbf{X}_{\text{MRND1}}$ exhibits qualitatively similar in-control trends to MND, with stronger complexity in RP1S SK at low d and more nuanced correlation effects for T^2 ; DI maintains a clear negative dependence on ρ . All quantitative summaries (means, quantiles, and dispersion measures) across d and ρ are reported in the Appendix (Table 4.2), which underpin these statements and ensure reproducibility of the claims.

Hadamard Product: Poisson - Normal $\mathbf{X}_{\text{MRND2}}$ displays rapid growth of most metrics with d and ρ , particularly for CONS SK ; DI decreases with ρ but less steeply than in MND, indicating partial stabilization due to the discrete-continuous mixture. The detailed values and percentiles that substantiate these observations are provided in the Appendix (Table 3.4).

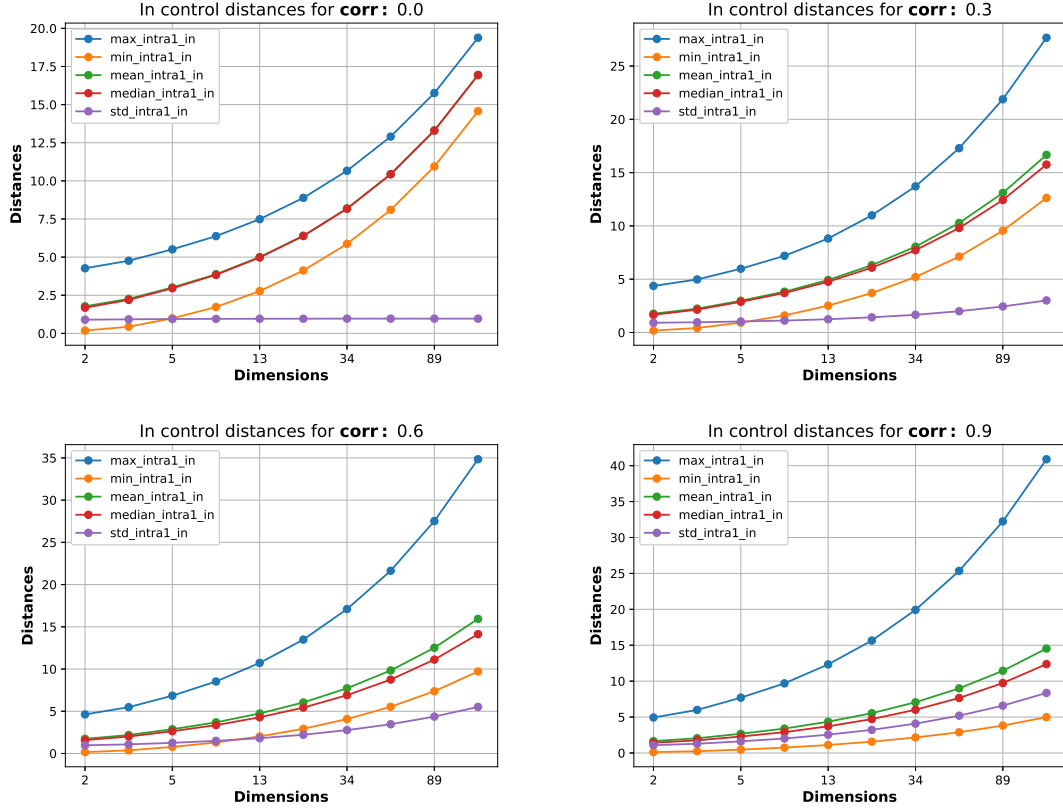


Fig. 3.12 Intra-cluster pairwise distances (Cluster 1) in Multivariate Normal Distribution across d and ρ , showing cohesion changes and heteroscedasticity with correlation.

3.6.2 Phase II: Mean Shift Detection

Multivariate Normal Distribution Under mean shifts, detection performance improves with dimension for T^2 and SK at low ρ but deteriorates as ρ increases; DI shows the weakest sensitivity at higher correlations, consistent with reduced separability under strong pairwise linear relationship. These trends are shown comprehensively in Figure 3.13.

Hadamard Product: Rayleigh - Normal $\mathbf{X}_{\text{MRND1}}$ detection results mirror the MND pattern at low ρ with pronounced dimensional gains in T^2 and CONS SK , but exhibit accelerated performance degradation as ρ increases; RP1S approach metrics remain less sensitive and DI underperforms at high correlation (Figure 3.14).

Hadamard Product: Poisson - Normal $\mathbf{X}_{\text{MRND2}}$ detection performance increases with dimension at low ρ for T^2 and CONS SK , while in RP1S metrics remain flatter;

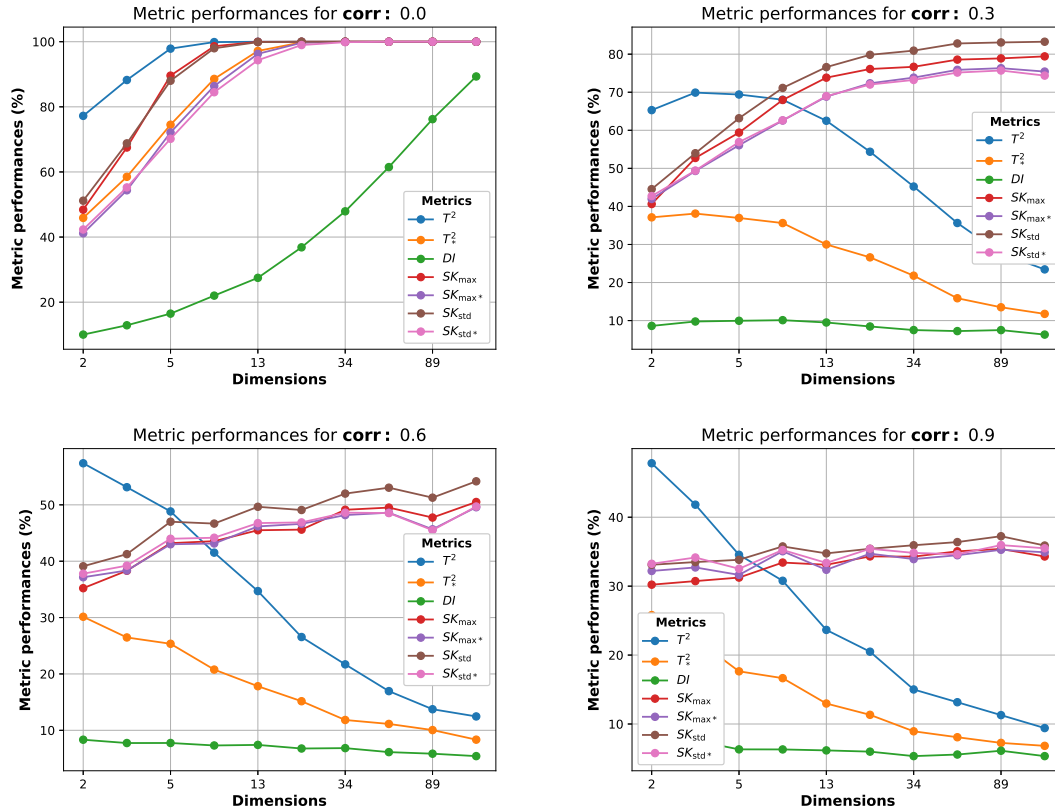


Fig. 3.13 Mean shift detection performance in Multivariate Normal Distribution across d and ρ , highlighting dimensional gains at low correlation and performance collapses in higher correlation.

all methods deteriorate quickly as ρ rises, with DI the poorest and CONS SK the most robust at low correlation (Figure 3.15).

3.7 Conclusions

The comprehensive evaluation of performance metrics across diverse operational scenarios, distributional frameworks, and monitoring methodologies enables the development of robust conclusions and offers valuable guidance for future methodological advancements and feature-based investigations. Across all experimental settings and distributional contexts examined, the observed result patterns were coherent and interpretable, supporting meaningful comparative analysis and facilitating practical recommendations.

The classical Hotelling T^2 statistic, particularly within the RP1S approach, exhibited strong detection capability in low-dimensional settings where the assumptions of

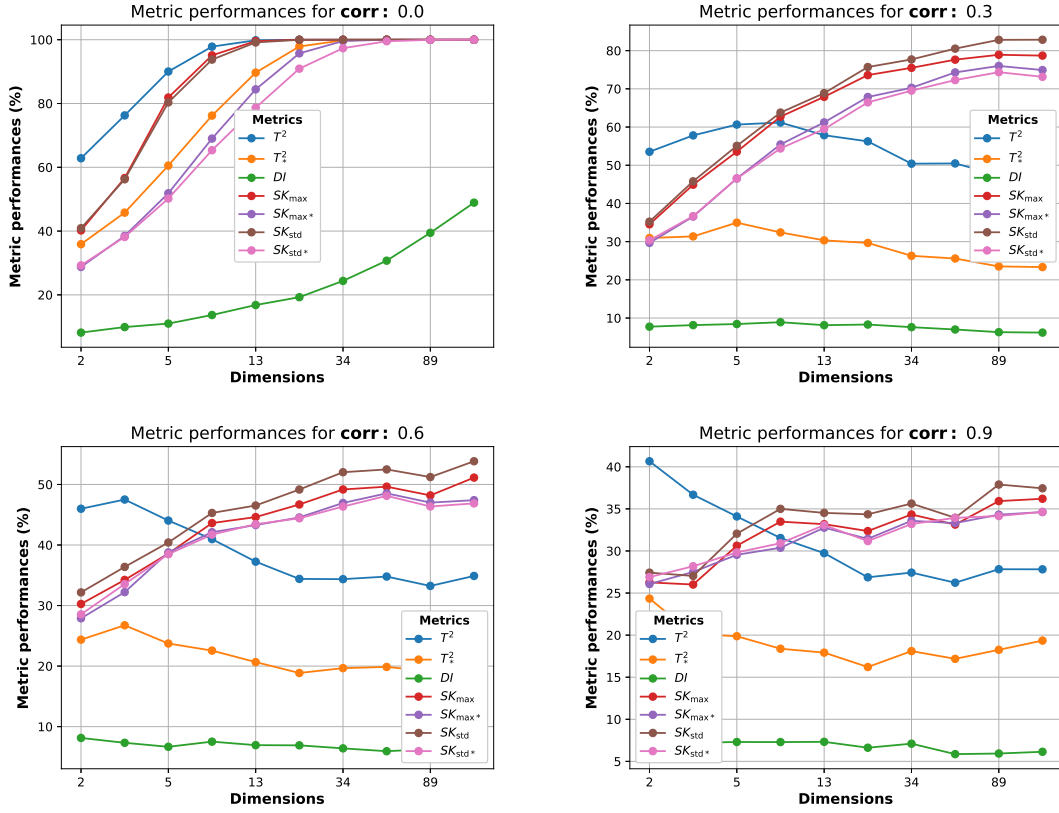


Fig. 3.14 Mean shift detection performance in Hadamard product of Rayleigh–Normal across d and ρ , showing rapid performance loss with correlation.

multivariate normality and reliable covariance estimation hold. These results reaffirm its effectiveness as a benchmark tool for monitoring processes in moderately correlated, low-dimensional feature spaces. However, its advantages decline substantially with increasing dimensionality and complex correlation structures, reflecting well-documented issues associated with covariance estimation instability and the curse of dimensionality.

In contrast, the proposed SK family of metrics, including their computational variations within both RP1S and CONS frameworks, demonstrate consistently high and robust detection performance across a wide range of operational conditions. Notably, these metrics maintain strong effectiveness in high-dimensional environments, where traditional measures such as Hotelling's T^2 and the DI tend to degrade in sensitivity. This robustness in high-dimensional scenarios presents a significant advantage for modern industrial and data-intensive applications characterized by large numbers of sensors, variables, or feature dimensions.

The DI , although conceptually attractive as a cluster separability index, is limited by the mathematical constraints of its formulation, particularly its dependence on

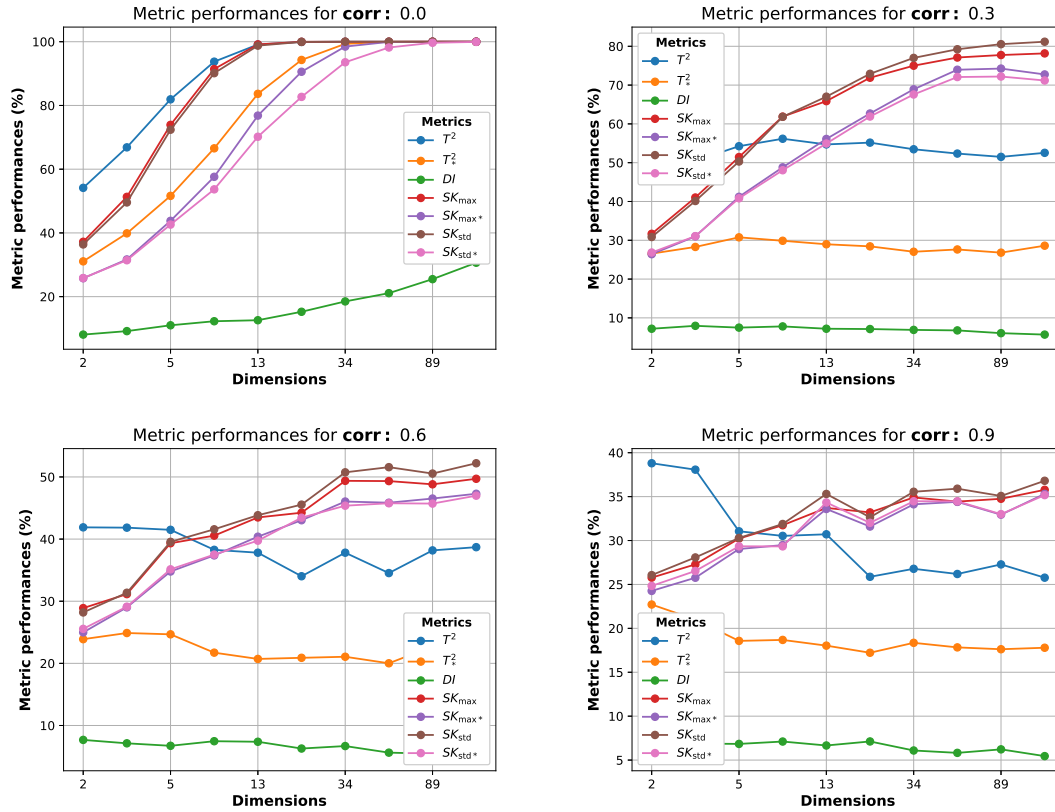


Fig. 3.15 Mean shift detection performance in Hadamard product of Poisson–Normal across d and ρ , illustrating early performance collapse with growing correlation.

the minimum inter-cluster distance. For small or subtle mean shifts, this minimum distance often approaches zero ($\min \approx 0$), leading to a loss of discriminatory capability and reduced ability to detect process deviations. Accordingly, DI appears suboptimal as a standalone separability metric for process monitoring when compared to SK and Hotelling's T^2 , both of which provide stronger sensitivity and stability under a broader range of conditions.

In the uncorrelated case, where inter-variable dependencies are weak or absent, most metrics display improved detection performance with increasing dimensionality, reflecting the additive informational gain of independent features. Conversely, as correlation strength increases from weak to strong, all metrics experience a marked decline in detection performance—a result consistent with the dimensional collapse effect, where redundant information from correlated variables confines the data to lower-dimensional subspaces, thereby limiting the discriminative capacity of distance- and covariance-based approaches.

When comparing detection under Hadamard product distributions (MND combined with MRD or MPD) to mean-shift detection in purely multivariate normal data, the overall performance patterns and qualitative dependencies on dimensionality and correlation remain consistent. Nonetheless, absolute performance levels are diminished in the hybrid distributional cases, reflecting added structural complexity, discretization effects, and distributional interactions introduced by multiplicative constructs. This indicates that processes characterized by non-Gaussian or mixed distributional behavior may require refined thresholding or adaptive calibration to sustain reliable detection rates.

A geometrically grounded perspective informed by concepts from high-dimensional geometry and manifold theory can provide deeper insight into these observed phenomena. While computational and statistical analyses remain essential, they may not fully account for the intrinsic structural features that govern separability and anomaly detection in high-dimensional spaces. The interplay between the curse and blessing of dimensionality is particularly influential, as noted by Aggarwal et al. [141], since data distributions typically occupy compact submanifolds within the high-dimensional ambient space. This geometric concentration fundamentally shapes the behavior of distance-based metrics and control chart performance. Future work should therefore investigate geometric invariants, manifold learning methods, and dimensionality reduction techniques to enhance both theoretical understanding and the practical design of robust monitoring schemes in complex, high-dimensional data environments.

3.8 Summary

This chapter provides a comprehensive examination of SPM methodologies for high-dimensional data streams, with particular emphasis on addressing the challenges posed by modern industrial and IoT-driven environments. The chapter explores the evolution from traditional univariate control charts to advanced multivariate monitoring frameworks capable of handling complex, high-velocity data streams characterized by evolving patterns, non-Gaussian distributions, and high dimensionality.

The foundational concepts of SPM are established through detailed coverage of classical control charting techniques, including Shewhart, CUSUM, and EWMA charts, which form the basis for detecting temporal variations and distinguishing between common cause and special cause variability. The chapter then transitions to multivariate monitoring contexts, introducing Hotelling's T^2 statistic as a fundamental tool for assessing deviations in multivariate processes. The mathematical formulation, distribu-

tional properties, and practical implementation of the T^2 control chart are thoroughly examined, along with its assumptions, limitations, and performance characteristics under varying dimensionality and correlation structures.

A significant portion of the chapter is dedicated to the recently introduced clustering-based monitoring approach, which leverages ML clustering techniques to enhance anomaly detection in multivariate settings. Unlike conventional control charts that rely on predefined statistical assumptions such as normality, cluster-based control charts exploit the intrinsic grouping structure of data by treating sequential samples as distinct clusters and evaluating their separability using appropriately defined distance-based indices. This approach proves particularly effective in complex processes where data exhibit nonlinear relationships, non-Gaussian distributions, or irregular patterns that render standard control charts less reliable.

The chapter introduced the DI as a classical cluster separability metric and presents a family of indices collectively designated as SK , which are specifically formulated to detect structural changes in data streams by leveraging pairwise distances, centroid separation, and boundary extremes. The SK family includes both reference-based variants (SK_{max} , SK_{std}) and consecutive comparison variants (SK_{max}^* , SK_{std}^*), providing flexibility for different monitoring contexts. The mathematical formulations, geometric interpretations, and theoretical justifications for these indices are comprehensively detailed, along with their integration into real-time control chart frameworks.

To rigorously evaluate the performance of the proposed methodologies, an extensive simulation study is conducted across multiple distributional frameworks, including pure MND and complex Hadamard product distributions formed by combining MND with MRD and MPD.

The simulation design incorporates systematic variations across three key parameters:

- Dimensionality: Increased from 2 to 144, following the Fibonacci sequence.
- Correlation: Varied across coefficients ranging from 0.0 to 0.9.
- Mean Shift Magnitude: Set to the 75th percentile of the in-control data distribution.

Two distinct monitoring approaches are implemented and compared: the RP1S approach, which leverages historical in-control data as a stable reference, and the CONS approach, which focuses on sequential sample comparisons without relying on a fixed reference.

The simulation results reveal several critical insights into metric behavior and detection performance. Hotelling's T^2 statistic demonstrates exceptional performance in low-dimensional spaces with moderate correlation but exhibits degraded sensitivity as dimensionality increases and correlation strengthens, reflecting limitations related to covariance matrix estimation and dimensional collapse. The newly proposed SK family of indices exhibits remarkable robustness and sustained effectiveness across high-dimensional spaces, often surpassing traditional metrics in detection power, particularly under low correlation conditions. The DI, while theoretically appealing, suffers from fundamental limitations due to its reliance on minimum inter-cluster distances, which approach zero for subtle shifts, rendering it less effective for practical process monitoring applications.

Performance trends across correlation levels reveal that all metrics benefit from low correlation environments where features provide complementary information, but suffer systematic performance degradation as correlation increases due to dimensional collapse and reduced discriminatory power. The Hadamard product distributions introduce additional complexity, with detection performance patterns remaining qualitatively similar to pure MND but with reduced absolute performance levels, reflecting the structural challenges posed by mixed continuous-discrete or asymmetric distributions.

The chapter concludes by synthesizing the findings and offering practical recommendations for metric selection based on operational context, dimensionality, and distributional characteristics. The results underscore the importance of adaptive, data-driven monitoring frameworks that can accommodate non-Gaussian distributions, high-dimensional feature spaces, and dynamic process behaviors. The discussion emphasizes the need for geometrically informed perspectives to fully understand high-dimensional phenomena and suggests future research directions including manifold learning, dimensionality reduction, and hybrid monitoring strategies that integrate statistical and ML approaches. Overall, the chapter establishes clustering-based separability indices, particularly the SK family, as promising and robust tools for real-time anomaly detection in modern, data-intensive industrial environments.

Statistic d	$\rho = 0.00$										$\rho = 0.30$										$\rho = 0.60$										$\rho = 0.90$												
	RPIS approach					CONS approach					RPIS approach					CONS approach					RPIS approach					CONS approach					RPIS approach					CONS approach							
	T^2	SK_{max}	SK_{std}	T^2_*	DI	SK_{max}^*	SK_{std}^*	T^2	SK_{max}	SK_{std}	T^2_*	DI	SK_{max}^*	SK_{std}^*	T^2	SK_{max}	SK_{std}	T^2_*	DI	SK_{max}^*	SK_{std}^*	T^2	SK_{max}	SK_{std}	T^2_*	DI	SK_{max}^*	SK_{std}^*	T^2	SK_{max}	SK_{std}	T^2_*	DI	SK_{max}^*	SK_{std}^*								
4*mean	3	3.0	2.97	0.57	5.96	0.09	6.36	1.23	2.99	3.38	0.65	5.97	0.09	6.69	1.29	3.02	4.12	0.81	6.13	0.07	7.55	1.49	3.02	5.09	1.09	6.06	0.04	8.39	1.81														
	5	4.96	2.58	0.44	9.95	0.18	7.12	1.22	4.97	3.28	0.57	9.97	0.16	7.8	1.35	4.99	4.32	0.79	9.98	0.12	9.12	1.69	5.03	5.77	1.21	10.11	0.06	10.69	2.27														
	8	7.93	2.3	0.34	16.04	0.27	8.31	1.24	7.97	3.34	0.52	15.88	0.23	9.42	1.47	8.01	4.79	0.84	16.04	0.16	11.49	2.03	8.0	6.67	1.38	15.9	0.08	13.45	2.82														
	144	144.24	1.64	0.08	288.76	0.75	46.39	2.31	144.21	9.18	0.99	288.7	0.47	67.29	7.42	144.05	18.66	2.94	288.09	0.29	87.47	14.11	144.46	32.36	6.59	288.72	0.13	109.19	22.7														
4*std	3	2.44	0.76	0.14	4.81	0.04	2.29	0.45	2.44	1.07	0.2	4.87	0.03	2.7	0.53	3.17	1.49	0.29	5.0	0.03	3.58	0.73	2.48	2.03	0.42	4.96	0.02	4.72	1.04														
	5	3.15	0.65	0.11	6.3	0.05	2.31	0.4	3.17	1.17	0.2	6.33	0.04	3.05	0.54	3.17	1.8	0.33	6.28	0.04	4.53	0.88	3.17	2.74	0.56	6.35	0.02	6.77	1.48														
	8	3.94	0.57	0.09	7.99	0.05	2.4	0.37	3.98	1.32	0.21	7.92	0.05	3.61	0.6	4.0	2.37	0.41	7.97	0.04	6.13	1.16	3.98	3.61	0.73	7.94	0.02	9.45	2.05														
	144	17.07	0.37	0.02	34.19	0.03	4.74	0.33	16.86	7.96	0.9	33.7	0.07	28.35	4.01	16.76	18.63	2.98	33.91	0.05	65.79	11.71	16.93	34.3	6.95	34.14	0.03	118.29	25.36														
4*min	3	0.01	1.63	0.27	0.0	0.01	1.84	0.36	0.0	1.64	0.33	0.0	0.0	2.19	0.4	0.0	1.81	0.34	0.01	0.0	2.32	0.44	0.0	2.07	0.46	0.02	0.0	2.58	0.55														
	5	0.11	1.5	0.22	0.16	0.02	2.21	0.36	0.1	1.58	0.26	0.21	0.03	1.96	0.37	0.1	1.7	0.27	0.1	0.01	2.62	0.5	0.11	1.95	0.41	0.16	0.01	2.81	0.58														
	8	0.31	1.41	0.18	1.02	0.09	2.49	0.39	0.52	1.47	0.2	0.95	0.07	3.03	0.45	0.26	1.61	0.27	0.8	0.04	2.75	0.43	0.47	2.13	0.45	1.15	0.02	3.16	0.66														
	144	87.88	1.18	0.05	170.4	0.59	30.09	1.3	90.94	1.35	0.09	185.76	0.26	30.5	1.97	91.7	1.69	0.19	167.53	0.15	24.6	1.97	82.97	2.17	0.43	173.89	0.06	13.13	2.78														
4*25%	3	1.21	2.41	0.47	2.42	0.07	4.68	0.91	1.22	2.6	0.5	2.39	0.06	4.74	0.91	1.22	3.03	0.6	2.47	0.05	5.08	0.99	1.22	3.61	0.78	2.42	0.03	5.07	1.08														
	5	2.65	2.1	0.36	5.31	0.15	5.43	0.92	2.66	2.43	0.42	5.33	0.13	5.68	0.98	2.69	2.99	0.55	5.36	0.09	6.01	1.1	2.73	3.76	0.79	5.45	0.05	5.88	1.22														
	8	5.06	1.88	0.28	10.16	0.24	6.6	0.98	5.06	2.36	0.37	10.07	0.19	6.97	1.07	5.09	3.08	0.54	10.15	0.13	7.4	1.27	5.05	4.02	0.84	10.08	0.07	6.82	1.4														
	144	132.28	1.35	0.07	265.14	0.73	43.1	2.08	132.62	3.6	0.38	265.51	0.42	49.84	5.04	132.36	6.0	0.94	264.52	0.25	46.73	7.17	132.73	9.04	1.83	264.77	0.11	30.28	6.11														
4*50%	3	2.34	2.8	0.54	4.74	0.09	5.99	1.15	2.36	3.13	0.6	4.71	0.08	6.16	1.19	2.38	3.74	0.73	4.82	0.07	6.65	1.3	2.36	4.57	0.98	4.8	0.04	6.94	1.48														
	5	4.31	2.43	0.41	8.71	0.18	6.81	1.16	4.35	2.99	0.52	8.67	0.16	7.18	1.24	4.29	3.86	0.71	8.7	0.12	7.9	1.44	4.38	5.02	1.06	8.84	0.06	8.55	1.78														
	8	7.28	2.17	0.32	14.72	0.28	8.03	1.19	7.32	3.01	0.47	14.51	0.23	8.69	1.34	7.39	4.13	0.72	14.78	0.16	9.64	1.67	7.38	5.64	1.17	14.66	0.08	10.27	2.11														
	144	143.35	1.53	0.08	287.19	0.76	46.22	2.29	143.68	6.53	0.69	287.17	0.47	58.5	6.33	143.47	12.4	1.94	286.27	0.29	62.98	9.79	143.89	20.48	4.16	287.85	0.13	64.32	13.08														
4*75%	3	4.15	3.37	0.65	8.09	0.12	7.67	1.48	4.09	3.9	0.75	8.22	0.11	8.04	1.55	4.13	4.82	0.95	8.37	0.09	9.03	1.78	4.17	6.01	1.28	8.2	0.05	10.33	2.23														
	5	6.58	2.92	0.5	13.2	0.21	8.46	1.44	6.57	3.83	0.66	13.22	0.19	9.26	1.59	6.6	5.16	0.95	13.15	0.14	10.89	2.0	6.62	7.02	1.47	13.3	0.08	13.33	2.81														
	8	10.13	2.6	0.39	20.48	0.31	9.71	1.45	10.2	3.97	0.62	20.38	0.26	10.98	1.71	10.21	5.85	1.02	20.5	0.18	13.6	2.38	10.25	8.31	1.72	20.31	0.09	16.82	3.55														
	144	155.28	1.83	0.09	310.91	0.77	49.42	2.52	155.18	11.95	1.29	310.43	0.52	74.39	8.32	155.11	24.8	3.87	310.44	0.32	103.5	16.47	155.37	43.01	8.74	311.12	0.14	142.27	29.24														
4*95%	3	7.89	4.44	0.85	15.36	0.16	10.71	2.06	7.76	5.43	1.05	15.78	0.15	11.86	2.32	7.77	7.08	1.38	16.12	0.12	14.55	2.92	7.95	9.0	1.9	15.78	0.07	18.09	3.92														
	5	10.96	3.84	0.66	22.19	0.26	11.37	1.96	11.05	5.61	0.96	22.14	0.23	13.57	2.38	11.1	7.9	1.45	22.13	0.18	18.33	3.47	11.1	11.07	2.3	22.54	0.1	24.57	5.28														
	8	15.23	3.41	0.51	31.23	0.36	12.68	1.91		15.43	0.93	30.75	0.31	16.3	2.62	15.4	1.66	31.13	0.23	23.9	4.35	15.38	13.75	2.82	30.74	0.12	32.74	7.01															
	144	173.59	2.38	0.12	347.64	0.8	54.47	2.89	172.8	25.17	2.8	346.93	0.58	123.39	15.08	172.38	56.11	8.78	345.53	0.38	217.94	37.18	173.14	102.57	20.96	347.05	0.18	354.96	74.61														
4*max	3	20.74	9.26	1.75	45.17	0.25	18.77	3.99	22.07	11.78	2.1	40.5	0.26	23.2	5.08	24.35	17.5	2.93	43.8	0.16	0.3	43.22	9.37	26.07	30.86	5.99	42.43	0.17	65.89	14.55													
	5	29.45	6.9	1.28	54.7	0.36	19.52	3.51	25.22	13.44	2.35	51.87	0.34	34.95	6.18	29.46	17.55	3.08	69.16	0.3	43.22	9.37	26.07	30.86	5.99	46.9	0.17	63.96	14.55														
	8	34.42	6.18	0.93	62.89	0.48	20.26	3.17	29.24	14.79	2.02	68.68	0.43	40.1	7.12	42.14	26.22	4.49	68.57	0.36	58.65	11.34	31.71	35.6	6.98	81.48	0.19	83.82	17.61														
	144	238.35	3.7	0.21	437.84	0.85	67.52	3.73	209.51	91.08	12.46	437.93	0.73	448.38	66.81	226.35	182.74	34.14	428.18	0.58	772.02	144.54	220.45	391.24	80.29	448.21	0.28	1296.94	283.96														

Statistic	d	$\rho = 0.00$						$\rho = 0.30$						$\rho = 0.60$						$\rho = 0.90$					
		RPIS approach			CONS approach			RPIS approach			CONS approach			RPIS approach			CONS approach			RPIS approach			CONS approach		
		T^2	SK_{max}	SK_{std}	T^2_*	DI	SK_{max}^*	SK_{std}^*	T^2	SK_{max}	SK_{std}	T^2_*	DI	SK_{max}^*	SK_{std}^*	T^2	SK_{max}	SK_{std}	T^2_*	DI	SK_{max}^*	SK_{std}^*	T^2	SK_{max}	SK_{std}
4*mean	3	2.99	3.9	0.78	5.97	0.07	10.32	2.06	2.99	4.35	0.87	6.09	0.07	10.63	2.12	2.93	5.08	1.02	5.94	0.06	11.27	2.27	2.97	6.07	1.27
	5	5.04	3.33	0.61	9.98	0.14	11.95	2.17	5.01	4.19	0.77	10.0	0.13	12.55	2.31	5.02	5.27	1.0	10.01	0.1	13.75	2.62	4.99	6.7	1.37
	8	7.96	2.98	0.49	15.95	0.21	14.32	2.35	7.95	4.16	0.7	15.94	0.19	15.3	2.58	8.0	5.71	1.02	16.01	0.15	17.49	3.15	8.01	7.65	1.53
	144	144.62	1.94	0.12	280.02	0.7	87.83	5.46	144.23	11.79	1.25	287.98	0.47	118.5	12.78	144.4	24.02	3.61	288.87	0.3	155.4	23.99	144.64	41.23	7.92
4*std	3	2.63	1.41	0.28	5.07	0.03	5.08	1.03	2.64	1.73	0.35	5.17	0.03	5.53	1.12	2.62	2.29	0.47	5.12	0.03	6.5	1.34	2.82	2.97	0.62
	5	3.45	1.14	0.21	6.62	0.04	5.09	0.96	3.45	1.88	0.35	6.59	0.04	5.87	1.12	3.48	2.66	0.51	6.7	0.04	7.5	1.5	3.85	3.67	0.75
	8	4.25	1.02	0.17	8.22	0.05	5.21	0.9	4.29	2.03	0.35	8.37	0.05	6	1.18	4.49	3.3	0.6	8.44	0.04	9.67	1.85	5.0	4.74	0.95
	144	18.24	0.61	0.04	35.54	0.04	10.0	0.94	18.86	11.16	1.23	36.33	0.07	43.37	6.12	24.27	25.49	3.87	41.79	0.06	101.04	17.61	46.97	44.98	8.58
4*min	3	0.0	1.81	0.33	0.01	0.0	2.15	0.46	0.01	1.79	0.01	0.0	0.0	2.24	0.45	0.0	1.98	0.39	0.0	0.0	2.49	0.51	0.0	1.97	0.44
	5	0.04	1.64	0.25	0.17	0.01	2.36	0.44	0.09	1.71	0.28	0.05	0.01	2.69	0.43	0.02	1.76	0.3	0.2	0.02	2.83	0.54	0.04	2.11	0.43
	8	0.23	1.5	0.21	0.51	0.06	3.09	0.48	0.39	1.53	0.23	0.62	0.05	2.72	0.44	0.26	1.71	0.26	1.03	0.03	3.86	0.65	0.38	1.98	0.43
	144	91.21	1.24	0.06	182.35	0.53	56.62	2.83	83.48	1.37	0.09	169.23	0.26	51.58	3.25	74.96	1.59	0.15	169.93	0.16	46.78	4.38	50.15	2.06	0.43
4*25%	3	1.14	2.88	0.57	2.33	0.05	6.65	1.32	1.14	3.09	0.61	2.37	0.05	6.75	1.34	1.09	3.41	0.69	2.25	0.04	6.83	1.36	1.04	3.94	0.83
	5	2.58	2.51	0.45	5.16	0.11	8.3	1.48	2.53	2.83	0.51	5.29	0.1	8.47	1.53	2.58	3.33	0.63	5.19	0.08	8.64	1.61	2.33	3.99	0.81
	8	4.86	2.23	0.36	9.91	0.18	10.59	1.7	4.83	2.69	0.45	9.9	0.16	10.66	1.76	4.76	3.33	0.59	9.98	0.12	11.07	1.94	4.49	4.19	0.84
	144	131.84	1.47	0.09	264.57	0.67	80.93	4.81	131.29	4.04	0.42	262.6	0.43	91.3	9.0	127.13	6.6	0.99	259.82	0.26	92.86	13.55	110.58	10.35	2.01
4*50%	3	2.27	3.56	0.71	4.6	0.07	9.24	1.83	2.26	3.89	0.77	4.69	0.06	9.39	1.86	2.2	4.49	0.9	4.57	0.06	9.57	1.92	2.14	5.25	1.1
	5	4.28	3.06	0.55	8.53	0.14	11.0	1.99	4.2	3.69	0.67	8.62	0.13	11.33	2.06	4.23	4.56	0.86	8.42	0.1	11.78	2.22	4.01	5.72	1.16
	8	7.17	2.72	0.44	14.6	0.21	13.55	2.21	7.16	3.6	0.6	14.46	0.19	13.98	2.32	7.17	4.74	0.85	14.46	0.14	14.93	2.65	6.87	6.36	1.27
	144	143.93	1.76	0.11	287.66	0.7	87.44	5.38	143.27	8.28	0.86	286.52	0.47	106.93	11.28	142.02	15.3	2.27	285.37	0.3	120.51	18.17	137.48	25.13	4.84
4*75%	3	4.06	4.56	0.9	8.17	0.09	12.79	2.55	4.02	5.16	1.03	8.34	0.09	13.09	2.61	3.94	6.13	1.23	8.14	0.07	13.79	2.77	3.94	7.35	1.54
	5	6.63	3.88	0.71	13.26	0.17	14.6	2.66	6.67	4.99	0.92	13.23	0.15	15.3	2.8	6.58	6.45	1.22	13.21	0.13	16.7	3.16	6.56	8.36	1.7
	8	10.19	3.47	0.57	20.3	0.25	17.16	2.83	10.15	5.06	0.85	20.37	0.22	18.36	3.1	10.21	7.1	1.27	20.38	0.17	20.77	3.74	10.3	9.7	1.93
	144	156.28	2.23	0.14	312.23	0.72	94.25	6.05	155.9	15.35	1.64	311.64	0.52	131.21	14.59	158.81	33.32	4.89	314.88	0.34	180.3	27.69	170.29	56.31	10.83
4*95%	3	8.1	6.64	1.33	15.97	0.13	19.98	4.02	8.1	7.79	1.56	16.26	0.12	21.16	4.27	8.06	9.61	1.92	15.77	0.11	23.84	4.82	8.41	11.96	2.51
	5	11.67	5.5	1.01	22.56	0.21	21.6	4.0	11.62	7.94	1.45	22.54	0.2	23.72	4.42	11.63	10.54	2.0	22.96	0.17	28.25	5.5	12.24	13.94	2.84
	8	16.0	4.96	0.82	31.51	0.3	23.91	4.0	16.02	8.14	1.38	31.74	0.27	27.56	4.79	16.41	12.17	2.19	31.9	0.22	36.45	6.79	17.44	16.86	3.37
	144	175.68	3.13	0.2	348.96	0.76	104.9	7.11	177.01	34.41	3.68	351.12	0.59	203.63	21.3	188.3	75.08	11.36	362.94	0.4	356.83	58.45	231.11	131.43	25.19
4*max	3	32.32	14.74	3.04	45.2	0.25	48.72	10.84	27.24	16.79	3.54	60.5	0.22	54.91	11.86	26.66	25.79	5.78	54.42	0.2	78.21	16.19	29.08	27.52	6.21
	5	36.12	11.13	2.26	73.95	0.35	47.62	8.86	34.93	20.72	4.28	67.39	0.32	67.03	14.66	37.58	24.92	4.57	58.05	0.29	77.24	16.94	48.81	31.72	7.1
	8	43.93	11.09	2.19	72.73	0.42	52.65	9.22	37.63	18.69	3.59	76.47	0.41	72.38	13.44	37.01	32.01	5.45	70.01	0.38	127.74	24.75	63.78	51.82	10.11
	144	228.69	5.58	0.42	465.28	0.81	143.99	9.82	230.37	106.84	12.89	452.29	0.72	580.57	78.74	258.4	254.81	36.65	509.18	0.6	1409.45	240.69	516.92	492.61	85.1

Table 3.3 Hadamard Product: Rayleigh - Normal for different correlation cases across

Table 3.4 Hadamard Product: Poisson - Normal for different correlation cases across dimensions

Statistic d	$\rho = 0.00$					$\rho = 0.30$					$\rho = 0.60$					$\rho = 0.90$													
	RPIS approach		CONS approach			RPIS approach		CONS approach			RPIS approach		CONS approach			RPIS approach		CONS approach											
	T^2	SK_{max}	SK_{sid}	T^2_*	DI	SK_{max}^*	SK_{sid}^*	T^2	SK_{max}	SK_{sid}	T^2_*	DI	SK_{max}^*	SK_{sid}^*	T^2	SK_{max}	SK_{sid}	T^2_*	DI	SK_{max}^*	SK_{sid}^*	T^2	SK_{max}	SK_{sid}	T^2_*	DI	SK_{max}^*	SK_{sid}^*	
4^{th} mean	3	3.0	5.62	1.15	6.08	0.06	21.34	4.36	3.03	6.41	1.31	6.1	0.06	21.79	4.45	2.99	7.57	1.56	5.92	0.05	22.58	4.65	3.0	8.9	1.88	6.03	0.04	24.34	5.15
	5	5.04	4.79	0.9	10.1	0.12	26.26	4.94	5.02	6.2	1.18	10.05	0.11	26.95	5.11	5.01	7.95	1.54	9.99	0.09	29.18	5.66	4.96	10.14	2.08	10.02	0.06	32.26	6.66
	8	7.99	4.27	0.73	16.09	0.19	32.9	5.65	8.0	6.2	1.08	16.09	0.17	34.45	6.01	8.05	8.92	1.63	16.06	0.14	38.89	7.14	8.04	12.03	2.41	16.02	0.09	45.04	9.11
	144	143.82	2.7	0.18	288.11	0.67	242.82	16.56	143.92	22.88	2.78	287.97	0.47	316.85	34.13	144.19	49.56	7.37	288.24	0.31	403.61	61.35	144.71	86.59	16.42	288.92	0.15	500.52	96.78
	144	142.74	2.31	0.16	286.17	0.67	241.27	16.3	142.47	14.16	1.46	286.06	0.47	288.85	30.33	141.06	28.31	4.2	284.63	0.3	327.65	48.83	136.03	49.64	9.35	278.92	0.15	347.84	65.93
4^{th} std	3	2.73	2.93	0.62	5.25	0.03	13.57	2.81	2.8	3.73	0.79	5.4	0.03	14.48	3.0	2.74	4.9	1.04	5.09	0.03	15.93	3.34	2.96	5.93	1.28	5.43	0.02	19.02	4.11
	5	3.57	2.5	0.48	6.67	0.04	13.89	2.7	3.55	3.9	0.76	6.75	0.04	15.19	2.98	3.69	5.48	1.09	6.83	0.04	19.33	3.88	3.96	7.5	1.55	7.23	0.03	25.36	5.38
	8	4.4	2.17	0.38	8.54	0.05	14.62	2.63	4.46	4.12	0.73	8.46	0.05	17.09	3.13	4.74	6.76	1.26	8.81	0.04	24.2	4.7	5.23	9.75	1.98	9.15	0.03	34.53	7.22
	144	18.91	1.27	0.09	35.91	0.04	30.34	3.17	20.39	25.12	2.76	37.57	0.07	110.14	15.6	28.64	58.64	8.89	47.33	0.06	242.21	42.33	52.15	103.8	19.74	78.15	0.03	425.51	86.43
	144	18.91	1.27	0.09	35.91	0.04	30.34	3.17	20.39	25.12	2.76	37.57	0.07	110.14	15.6	28.64	58.64	8.89	47.33	0.06	242.21	42.33	52.15	103.8	19.74	78.15	0.03	425.51	86.43
4^{th} min	3	0.0	1.82	0.36	0.01	0.0	2.74	0.53	0.0	1.82	0.38	0.01	0.0	2.86	0.52	0.0	2.0	0.39	0.0	0.0	2.17	0.42	0.01	2.12	0.46	0.0	0.0	2.71	0.58
	5	0.08	1.62	0.28	0.06	0.01	2.69	0.45	0.04	1.73	0.31	0.2	0.01	3.45	0.64	0.04	1.83	0.33	0.15	0.01	2.99	0.53	0.06	2.04	0.36	0.17	0.01	4.06	0.74
	8	0.36	1.56	0.24	1.32	0.02	6.3	1.0	0.38	1.53	0.25	0.51	0.04	4.98	0.78	0.18	1.76	0.29	0.99	0.02	6.4	1.1	0.3	1.97	0.39	0.69	0.01	4.97	0.88
	144	86.03	1.32	0.06	171.79	0.5	158.08	7.12	85.22	1.49	0.11	168.28	0.26	140.08	9.98	69.17	1.62	0.18	147.44	0.16	126.53	13.06	38.3	2.03	0.4	106.52	0.06	80.52	13.98
	144	86.03	1.32	0.06	171.79	0.5	158.08	7.12	85.22	1.49	0.11	168.28	0.26	140.08	9.98	69.17	1.62	0.18	147.44	0.16	126.53	13.06	38.3	2.03	0.4	106.52	0.06	80.52	13.98
$4^{25\%}$	3	1.12	3.5	0.71	2.34	0.04	11.85	2.4	1.08	3.76	0.76	2.33	0.04	11.78	2.39	1.1	4.16	0.84	2.3	0.03	11.86	2.41	1.06	4.75	0.99	2.24	0.02	11.78	2.46
	5	2.47	3.01	0.56	5.25	0.09	16.35	3.03	2.47	3.43	0.64	5.16	0.08	16.27	3.04	2.43	4.03	0.77	5.15	0.07	16.4	3.13	2.22	4.89	0.99	4.94	0.05	16.12	3.26
	8	4.79	2.68	0.46	9.88	0.15	22.49	3.79	4.79	3.26	0.56	9.95	0.14	22.52	3.84	4.7	4.16	0.75	9.77	0.11	22.81	4.06	4.34	5.15	1.02	9.41	0.06	22.6	4.49
	144	130.86	1.74	0.12	263.31	0.64	221.62	14.3	129.79	6.04	0.62	262.16	0.43	247.37	24.23	123.97	10.74	1.56	254.49	0.27	256.77	36.52	106.65	17.52	3.32	233.11	0.13	239.41	44.74
	144	142.74	2.31	0.16	286.17	0.67	241.27	16.3	142.47	14.16	1.46	286.06	0.47	288.85	30.33	141.06	28.31	4.2	284.63	0.3	327.65	48.83	136.03	49.64	9.35	278.92	0.15	347.84	65.93
$4^{50\%}$	3	2.22	4.79	0.97	4.67	0.05	18.08	3.64	2.25	5.34	1.08	4.61	0.05	18.03	3.66	2.23	6.11	1.25	4.5	0.05	18.26	3.74	2.15	7.12	1.5	4.49	0.04	18.62	3.91
	5	4.21	4.07	0.76	8.67	0.12	23.39	4.36	4.18	5.03	0.94	8.65	0.11	23.51	4.42	4.06	6.35	1.21	8.47	0.09	24.02	4.62	3.88	7.81	1.59	8.22	0.06	24.53	5.02
	8	7.14	3.67	0.63	14.49	0.19	30.2	5.16	7.1	4.95	0.85	14.61	0.17	30.79	5.31	7.01	6.85	1.24	14.44	0.13	32.38	5.88	6.83	9.04	1.8	14.22	0.08	34.34	6.86
	144	142.74	2.31	0.16	286.17	0.67	241.27	16.3	142.47	14.16	1.46	286.06	0.47	288.85	30.33	141.06	28.31	4.2	284.63	0.3	327.65	48.83	136.03	49.64	9.35	278.92	0.15	347.84	65.93
	144	142.74	2.31	0.16	286.17	0.67	241.27	16.3	142.47	14.16	1.46	286.06	0.47	288.85	30.33	141.06	28.31	4.2	284.63	0.3	327.65	48.83	136.03	49.64	9.35	278.92	0.15	347.84	65.93
$4^{75\%}$	3	4.01	6.86	1.4	8.23	0.07	27.09	5.54	4.08	7.93	1.62	8.29	0.07	27.65	5.63	3.98	9.43	1.94	8.1	0.07	28.18	5.76	3.97	11.18	2.35	8.11	0.05	30.57	6.45
	5	6.67	5.83	1.1	13.43	0.14	33.04	6.21	6.66	7.74	1.46	13.26	0.14	33.79	6.38	6.58	10.17	1.95	13.17	0.12	35.76	6.93	6.48	13.04	2.68	13.11	0.08	40.04	8.22
	8	10.22	5.22	0.89	20.56	0.22	40.41	6.93	10.3	7.86	1.37	20.6	0.2	42.11	7.36	10.32	11.41	2.09	20.34	0.16	47.53	8.7	10.42	15.55	3.1	20.55	0.1	55.69	11.18
	144	155.94	3.23	0.22	311.29	0.7	262.05	18.48	156.46	30.83	3.24	311.92	0.52	349.87	39.15	160.93	65.97	9.64	317.1	0.35	459.61	69.98	174.13	115.15	21.71	334.34	0.17	602.62	115.38
	144	155.94	3.23	0.22	311.29	0.7	262.05	18.48	156.46	30.83	3.24	311.92	0.52	349.87	39.15	160.93	65.97	9.64	317.1	0.35	459.61	69.98	174.13	115.15	21.71	334.34	0.17	602.62	115.38
$4^{95\%}$	3	8.26	11.47	2.39	16.19	0.11	47.79	9.84	8.44	13.63	2.84	16.61	0.11	49.26	10.19	8.26	17.13	3.55	15.86	0.1	54.06	11.27	8.47	20.58	4.38	16.86	0.08	62.92	13.43
	5	11.86	9.75	1.86	22.94	0.19	52.37	9.98	11.89	13.78	2.67	22.9	0.18	55.85	10.7	12.07	18.74	3.65	22.99	0.16	67.52	13.31	12.79	24.83	5.13	24.19	0.11	80.2	16.89
	8	16.33	8.55	1.49	32.31	0.27	60.4	10.67	16.55	14.39	2.54	32.41	0.26	67.52	12.08	17.09	22.58	4.15	32.55	0.21	86.75	16.29	17.94	34.83	6.43	33.65	0.14	112.78	26.49
	144	176.57	5.28	0.37	350.56	0.74	295.19	22.15	179.87	72.84	7.92	352.49	0.59	535.53	64.28	195.71	167.17	25.02	372.07	0.41	886.72	142.61	242.62	294.74	56.28	434.01	0.21	1316.6	261.22
	144	176.57	5.28	0.37	350.56	0.74	295.19	22.15	179.87	72.84	7.92	352.49	0.59	535.53	64.28	195.71	167.17	25.02	372.07	0.41	886.72	142.61	242.62	294.74	56.28	434.01	0.21	1316.6	261.22
4^{th} max	3	36.75	28.78	5.95	55.86	0.23	145.9	28.64	29.84	44.85	8.86	65.41	0.24	152.1	30.96	27.18	58.41	13.1	53.32	0.19	175.22	37.1	37.69	59.48	13.15	68.74	0.18	230.01	60.31
	5	36.36	25.37	6.13	59.74	0.23	141.38	28.65	31.31	47.42	9.89	68.16	0.29	176.18	33.7	36.54	58.6	17.41	57.73	0.26	211.36	44.08	43.61	72.91	16.01	65.46	0.21	322.67	64.87
	8	36.97	31.66	5.21	68.12	0.42	127.69	27.62	40.97	34.46	7.09	68.83	0.38	192.13	32.08	45.73	79.44	13.22	58.72	0.32	262.17	56.94	54.94	95.88	22.47	92.81	0.23	366.66	78.12
	144	243.2	11.27	0.88	454.11	0.81	392.02	34.67	243.93	271.64	31.05	466.98	0.75	1300.92	174.14	334.17	538.91	89.71	526.44	0.58	4207.2	863.26	459.66	1222.46	219.7	757.19	0.33	4574.02	1036.64
	144	243.2	11.27	0.88	454.11	0.81	392.02	34.67	243.93	271.64	31.05	466.98	0.75	1300.92	174.14	334.17	538.91	89.71	526.44	0.58	4207.2	863.26	459.66	1222.46	219.7	757.19	0.33	4574.	

Chapter 4

Monitoring Data Streams with Separability Indices

4.1 Introduction

This chapter builds upon the novel clustering validity metric developed in this thesis, which was systematically assessed through the extended evaluation. While we previously proposed this innovative metric for detecting changes in high-dimensional streaming data, the present study rigorously compares its performance against fundamental and widely used CVIs. Through comprehensive empirical testing across simulated and practical scenarios, we elucidate the relative strengths, limitations, and applicability of these metrics for real-time anomaly detection and process monitoring tasks.

The fundamental trade-off between simplicity and complexity in RTM metric development depends on three key system characteristics: latency constraints, computational resource availability, and required detection accuracy. Simple metrics, such as moving averages or threshold-based approaches, are computationally lightweight and operationally fast, making them particularly well-suited for high-frequency, low-latency applications where computational performance is critical. However, these simpler approaches may fail to capture subtle patterns and can result in elevated rates of false positives or false negatives. Conversely, more sophisticated metrics, such as ML-based anomaly detection algorithms or advanced statistical methods, offer substantially higher accuracy and the capability to capture intricate behavioral patterns and complex relationships. Nevertheless, these approaches impose higher computational demands and introduce increased latency, rendering them more appropriate for systems where detection precision takes precedence over immediate performance considerations. A hybrid monitoring strategy, strategically combining simple metrics for real-time alerting

with complex analytical models for deeper post-event analysis, frequently provides a balanced and effective solution, leveraging the complementary strengths of both methodologies. Ultimately, the optimal choice depends critically on the application's specific operational requirements, including error tolerance thresholds, data volume characteristics, and the criticality of real-time performance constraints.

In the previous chapter, a recently introduced family of indices was presented as an analytical tool designed to enhance the detection of abrupt changes in data by leveraging the fundamental statistical principles underlying control charts. This metric was systematically compared to the well-established multivariate Hotelling T^2 statistic, which is extensively utilized in multivariate normal distribution analysis and process monitoring applications. The comparative analysis aimed to rigorously validate the robustness and superior detection capability of the introduced metric in identifying sudden process shifts, thereby providing a more reliable and sensitive tool for change detection in complex operational environments. Moreover, the analysis was extended beyond the multivariate normal distribution framework, ensuring that the metric also performs effectively across various alternative random variable distributions, including non-Gaussian and mixed distributional contexts. The comprehensive simulation results suggest that the presented metric demonstrates considerable promise as a versatile, robust, and broadly applicable tool for change detection across diverse statistical contexts and operational scenarios.

In operational industrial systems, continuous process monitoring is critical to ensure both product quality and operational efficiency. MSPM has traditionally served as the foundational framework for process monitoring, employing control charts to detect shifts, trends, or outliers in multivariate data. However, with the increasing prevalence of high-velocity, continuous data flows characteristic of modern industrial and IoT-enabled systems, traditional MSPM techniques encounter significant challenges related to real-time analysis capabilities, computational scalability, and adaptive monitoring. The most substantial challenge concerns the distributional properties of the data flow. In the majority of practical cases, the actual underlying distribution is unknown and, furthermore, is difficult or computationally prohibitive to identify accurately. In this chapter, a comprehensive set of monitoring schemes is presented, exploiting a class of indices specifically designed to quantify distances among clusters, commonly referred to as cluster separability measures. By strategically leveraging CVIs, such as the Silhouette Index (SHI), Dunn Index (DI), Calinski-Harabasz Index (CHI), and Davies-Bouldin Index (DBI), it becomes possible to dynamically monitor the quality of clusters in real-time [142], thereby enabling the identification of emerging patterns

and deviations that might not be immediately evident when analyzing individual observations in isolation. This strategic combination of MSPM principles with Cluster Validity concepts provides a powerful and synergistic approach for anomaly detection, substantially enhancing the identification of special-cause variations in rapidly evolving data streams. Through detailed case studies utilizing simulated data representative of real-world operational conditions, the potential of this integrated approach to enhance process stability, minimize unplanned downtime, and optimize resource utilization in high-frequency data environments is systematically demonstrated. In addition, the efficacy and comparative performance of various CVIs, evaluated through their respective control chart implementations, are rigorously compared, thereby shedding light on their relative effectiveness, sensitivity, and practical utility in process monitoring applications.

In today's rapidly changing industrial and operational settings, continuous monitoring of process activities is essential to maintain both product quality and operational efficiency. Multivariate Statistical Process Control (MSPC) extends traditional univariate process control methods by incorporating multiple correlated variables simultaneously into the monitoring and regulation framework of a process. In contrast to univariate Statistical Process Control (SPC), which monitors individual process variables separately and independently, MSPC explicitly recognizes that real-world industrial processes often involve complex interactions and interdependencies between various operational factors and measured variables. By employing multivariate analytical techniques such as MSPM enables the simultaneous detection of process shifts or anomalies across several correlated variables. This holistic and integrated approach allows for a more comprehensive and accurate understanding of process behavior, making it possible to identify and correct operational issues that might be overlooked or missed when analyzing variables in isolation. MSPC is particularly valuable and widely adopted in industries such as manufacturing, chemical processing, and pharmaceutical production, where the interactions and interdependencies between process variables significantly impact product quality, yield, and overall process efficiency.

Although MSPC and MSPM are built upon similar foundational principles, both utilizing multivariate statistical methods to evaluate, monitor, and enhance process operations, the term MSPM has gained increasing prominence and widespread acceptance in the realm of statistical process management in recent years. This terminological shift is largely attributable to MSPM's broader and more inclusive conceptual scope, which integrates the fundamental principles of MSPC while placing substantially greater emphasis on the real-time, continuous observation and monitoring of process performance

over extended operational periods. MSPM is primarily concerned with identifying potential process deviations and emerging problems proactively before they escalate into major quality issues or process failures, thereby allowing for timely preventive measures and interventions that boost the overall sustainability and effectiveness of continuous process enhancement initiatives. As industrial systems progress and manufacturing processes grow increasingly intricate, interconnected, and data-intensive, the demand for a more adaptable, flexible, and forward-thinking monitoring strategy has rendered MSPM the preferred and dominant framework for ensuring product quality, process reliability, and operational excellence. As a result, MSPM has become the most prevalent and widely adopted terminology in contemporary industrial practice, highlighting the growing significance and critical importance of continuous observation, RTM, and early detection capabilities in modern industrial and manufacturing operations.

MSPM has long been regarded as the gold standard and established benchmark for detecting process shifts, trends, or outliers in manufacturing and operational processes across diverse industries. MSPM typically relies on control charts [143] that provide statistical signals when a process is deemed to be operating in an out-of-control state by systematically comparing observed data to pre-established control limits derived from historical in-control data [42]. However, traditional MSPM methodologies encounter significant limitations and practical challenges when applied to data streams, which are characterized by continuous, high-speed data flows that are becoming increasingly common and prevalent in modern industrial systems, particularly those incorporating real-time sensor networks and IoT infrastructures. These data streams present substantial challenges in terms of real-time analysis requirements, computational scalability constraints, and the detection of subtle, gradual changes in process behavior over extended time horizons.

Data streams are fundamentally characterized by their large data volume, high acquisition frequency, and real-time operational nature [144]. In such dynamic and demanding environments, the traditional MSPM framework, which was originally designed and optimized for discrete data sets or batch manufacturing processes, struggles to adequately handle the dynamic, evolving nature of continuous data streams. Consequently, there exists a growing and pressing need for advanced, adaptive methodological approaches that can effectively accommodate the evolving characteristics of the process, respond to non-stationary distributional properties, and provide timely, actionable insights in real-time operational contexts.

Assessment of the quality and validity of clusters formed during the monitoring process can be systematically accomplished by utilizing CVIs, including the SHI, DI,

CHI, and DBI. These indices provide valuable quantitative metrics for rigorously assessing how well data points are grouped and clustered, thereby aiding in anomaly detection and the identification of special-cause variations that may require immediate operational intervention. The SHI, for instance, quantitatively measures how similar an individual data object is to its own assigned cluster compared to other neighboring clusters, thereby helping to identify well-separated and internally cohesive clusters. Conversely, the Dunn Index (DI) focuses on maximizing the inter-cluster distance while simultaneously minimizing the intra-cluster distance, which proves particularly useful for detecting outliers and evaluating cluster compactness. The CHI evaluates the ratio of the sum of between-cluster dispersion to within-cluster dispersion, offering a robust method to assess the relative compactness and separation quality of clusters. The DBI, similarly, calculates the average similarity between each cluster and the cluster most similar to it, thereby providing a comprehensive measure of cluster separation and internal compactness.

These validity indices are particularly useful and widely applicable in high-dimensional datasets, as they offer a systematic and quantitative way to evaluate clustering outcomes without requiring prior knowledge of the true underlying structure of the data. They also play a crucial and indispensable role in anomaly detection applications, as poor cluster quality, reflected in degraded validity index values, often indicates the presence of data points that deviate significantly from normal operational patterns, thereby warranting further investigation and potential corrective action.

However, a primary and well-documented limitation of these validity indices is their inherent inability to accurately assess clusters of arbitrary, irregular shapes, especially when the underlying data distribution is non-convex, exhibits highly irregular structures, or contains nested cluster formations. This limitation arises fundamentally because these indices typically select a representative point, such as the cluster centroid, from each cluster to measure the distances between these representative points, and subsequently compute additional statistical parameters, such as variance or dispersion, based on them. As a result, they are less effective and reliable at distinguishing between clusters that exhibit highly irregular, elongated, or geometrically complex shapes. For example, when clusters are highly non-spherical, when nested clusters are present, or when data distributions exhibit complex manifold structures, traditional validity indices may fail to capture the true geometric and topological nature of the data [145]. Therefore, in practical cases where clusters do not exhibit simple, convex geometric shapes, alternative methodological approaches, such as density-based clustering algorithms, model-based clustering approaches, or manifold learning

techniques, may be more appropriate and effective for achieving an accurate assessment of cluster quality and process behavior.

4.2 Motivation

The exponential growth of data-intensive industrial systems, coupled with the proliferation of IoT devices and sensor networks, has fundamentally transformed the landscape of process monitoring and quality control. Modern manufacturing and operational environments generate continuous, high-velocity data streams that demand real-time analysis, adaptive decision-making, and rapid anomaly detection capabilities. Traditional MSPM methodologies, while theoretically sound and historically effective, were designed primarily for discrete batch processes and static distributional assumptions, rendering them increasingly inadequate for contemporary data-driven environments characterized by dynamic distributions, high dimensionality, and unknown or evolving statistical properties.

The central challenge confronting modern process monitoring lies in balancing competing operational demands: the need for computationally efficient, low-latency detection mechanisms that can operate continuously without overwhelming system resources, and simultaneously, the requirement for sophisticated analytical methods capable of capturing subtle process deviations, complex inter-variable relationships, and non-Gaussian distributional behaviors. Simple threshold-based metrics offer rapid computation and minimal latency but sacrifice detection sensitivity and robustness, leading to elevated false alarm rates or missed anomalies. Conversely, complex ML-based approaches provide superior accuracy and pattern recognition capabilities but impose substantial computational burdens, increased latency, and scalability limitations that may render them impractical for high-frequency streaming environments.

This fundamental trade-off between simplicity and sophistication motivates the development of innovative monitoring frameworks that strategically integrate the statistical rigor and interpretability of traditional MSPM with the adaptive flexibility and pattern recognition capabilities of modern ML methodologies. The family of cluster separability indices, presented in the previous chapter, demonstrated promising performance in detecting abrupt mean shifts across diverse distributional contexts, including multivariate normal distributions and complex Hadamard product distributions. These indices exhibited remarkable robustness in high-dimensional settings where conventional metrics experienced severe performance degradation, suggesting their potential as versatile tools for real-time process monitoring.

However, the practical applicability and operational effectiveness of cluster separability indices extend beyond their theoretical properties and simulation-based validation. A critical gap remains in systematically evaluating and comparing a broader spectrum of CVIs, including the SHI, DI, CHI, and DBI, within the specific context of real-time process monitoring and control chart frameworks. While these indices have been extensively studied in the ML and clustering literature for evaluating static clustering solutions, their integration into dynamic, sequential monitoring schemes for data streams presents unique challenges and opportunities that have not been thoroughly investigated.

Furthermore, the majority of real-world industrial processes operate under conditions where the underlying data distribution is either unknown, difficult to characterize accurately, or subject to gradual evolution over time. This distributional uncertainty poses significant challenges for traditional MSPM methods that rely heavily on parametric assumptions, particularly normality, and stable covariance structures. The increasing prevalence of complex, non-convex cluster geometries, nested cluster formations, and irregular data manifolds in modern high-dimensional datasets further exacerbates these challenges, as classical validity indices often fail to accurately assess cluster quality under such conditions.

These practical realities motivate the need for comprehensive comparative studies that rigorously evaluate the performance, sensitivity, computational efficiency, and robustness of various CVIs when deployed as RTM metrics. Specifically, there exists a pressing need to understand how different CVIs respond to varying degrees of cluster separation, different cluster shapes and densities, high-dimensional feature spaces, and evolving distributional properties characteristic of real-world data streams. Such systematic evaluation enables practitioners to make informed decisions regarding metric selection based on their specific operational requirements, including error tolerance, computational constraints, data characteristics, and the criticality of timely anomaly detection.

4.3 Materials and Methods

This section presents the methodology and materials employed to bridge MSPM principles with CVIs. By integrating MSPM's rigorous statistical frameworks with the versatile and data-driven clustering techniques commonly used in ML, the presented approach aims to enhance process monitoring and anomaly detection in complex, high-dimensional data environments. The section details the theoretical foundations, key

tools, and procedural steps that underpin this integration, providing a comprehensive basis for the methods evaluated in subsequent analyses.

4.3.1 Core Principles and Implementation Phases

MSPM is a methodology dedicated to the oversight and control of processes where multiple, correlated variables must be jointly considered to ensure process quality and stability. Unlike traditional univariate SPM, pioneered by Shewhart in the 1920s [146, 147], MSPM expands the scope to multivariate data, providing the foundation for robust control in industries marked by complex variable interdependencies.

Central to MSPM are multivariate control charts, such as Hotelling's T^2 [148], Multivariate CUSUM, and PCA-based control charts, which collectively visualize shifts or variability in process states that might remain undetected when variables are considered in isolation. These tools allow practitioners to monitor the joint behavior of process variables and efficiently detect deviations from normal operating conditions.

A typical MSPM implementation unfolds in two principal phases [149]: the first, stabilization, seeks to establish a baseline for statistical control through representative sampling and assessment of the process for stability and capability. It involves verifying essential assumptions, such as multivariate normality and independence, to maintain the validity of subsequent monitoring. The second, monitoring, involves continuous surveillance for process variations, facilitating prompt detection of anomalies or faults that could signal quality concerns.

By combining statistical rigor with modern data-driven techniques, MSPM underpins high-quality process management in settings ranging from manufacturing to service industries, serving as a cornerstone for both traditional and emerging data analytics frameworks.

4.3.2 Clustering Data Streams in Real-Time

Clustering refers to the task of partitioning a set of objects into groups, or clusters, in such a way that the similarity among objects within the same group is maximized, while the similarity between objects in different groups is minimized. In the context of data streams, the clustering problem becomes significantly more challenging. It involves maintaining a continuously accurate and meaningful clustering of the data points observed thus far, while operating under strict constraints on memory usage and computational time.

This challenge arises from the nature of data streams [121], where data points arrive in a continuous, potentially unbounded flow and must be processed in real time. As a result, traditional batch clustering methods have become impractical. Instead, stream clustering demands incremental and adaptive approaches that can dynamically update existing cluster structures as new data arrive. These methods must be capable of preserving the overall coherence of the clusters while efficiently incorporating the incoming data.

4.3.3 Clustering Validity Indices

To rigorously assess the quality of clusters and ensure that the clustering process yields meaningful, interpretable, and statistically significant groupings, several CVIs have been proposed and extensively investigated in the literature. These indices provide quantitative measures that evaluate how well the data points are grouped within individual clusters, reflecting internal cohesion, and how well-separated the resulting clusters are from each other, reflecting external separation. Commonly employed and widely adopted indices include:

- **Silhouette Index:** This index measures how close each individual data point is to its own assigned cluster relative to neighboring clusters [9]. A high SHI score indicates that the data points are well-clustered with strong internal cohesion and clear separation from other clusters, whereas a low score suggests that the data points are poorly clustered, potentially misassigned, or located in overlapping regions between clusters. The SHI score for an individual point quantifies how appropriately that point has been assigned to its cluster and is formally given by:

$$S_i = \frac{b_i - a_i}{\max(a_i, b_i)}$$

where:

- a_i represents the average distance between point $\mathbf{a}_i$ and all other points residing in the same cluster, thereby quantifying intra-cluster cohesion or compactness.
- b_i denotes the minimum average distance from point $\mathbf{a}_i$ to any point in a different, neighboring cluster, thereby quantifying inter-cluster separation or distinctness.

The overall SHI for the entire clustering solution, aggregated across all data points, is calculated as:

$$SHI = \frac{1}{n} \sum_{i=1}^n S_i$$

where n denotes the total number of points across all clusters. Higher SHI values indicate superior clustering quality characterized by well-defined, cohesive, and clearly separated clusters.

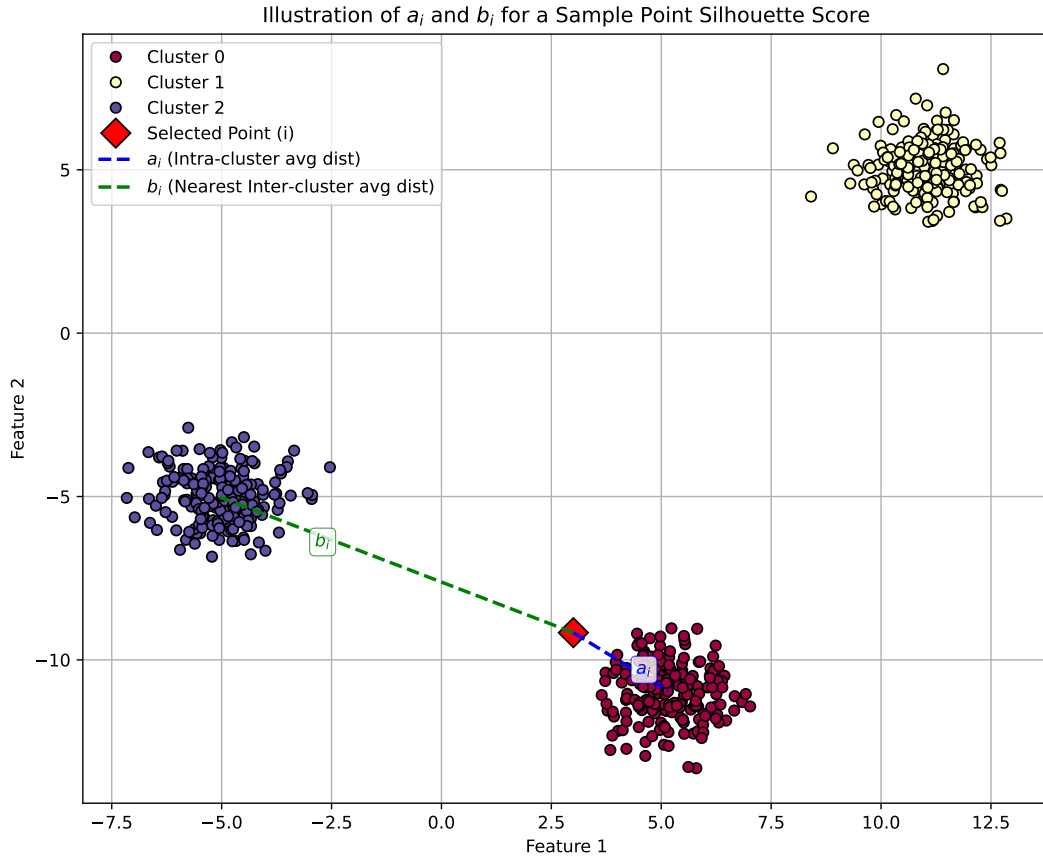


Fig. 4.1 Visualization of a_i and b_i for a selected data point (represented by the red diamond marker). The blue dashed line (a_i) represents the mean intra-cluster distance (cohesion), while the green dashed line (b_i) shows the mean nearest-cluster distance (separation).

- **Dunn Index:** As previously discussed in detail in the preceding chapter (see subsection 3.6), the DI evaluates cluster separability by considering the ratio of the minimum distance between distinct clusters to the maximum distance within individual clusters (see Equation 3.4.2). The DI assesses clustering quality by identifying and rewarding cluster configurations that are simultaneously compact, exhibiting small internal diameters, and well-separated, exhibiting large inter-cluster distances (see Figure 3.4 for geometric interpretation).
- **Calinski-Harabasz Index:** Also widely known as the variance ratio criterion, this index evaluates the ratio of the between-cluster dispersion, which quantifies cluster separation, to the within-cluster dispersion, which quantifies cluster compactness and internal homogeneity. Higher values of the CHI Index suggest superior clustering quality characterized by well-separated and internally cohesive clusters [150]. The CHI is formally defined as:

$$CHI = \frac{Tr(\mathbf{B}_k)}{Tr(\mathbf{W}_k)} \cdot \frac{n - k}{k - 1}$$

where:

- $Tr(\mathbf{B}_k)$ denotes the trace of the between-cluster dispersion matrix, which is computed as the sum of squared Euclidean distances between each cluster's centroid and the overall data centroid, thereby quantifying the degree of cluster separation and spatial distribution.
- $Tr(\mathbf{W}_k)$ denotes the trace of the within-cluster dispersion matrix, which is computed as the sum of squared Euclidean distances between individual points in each cluster and their respective cluster centroids, thereby quantifying cluster compactness and internal variability.
- n represents the total number of data points across all clusters in the dataset.
- k represents the number of clusters in the clustering solution being evaluated.

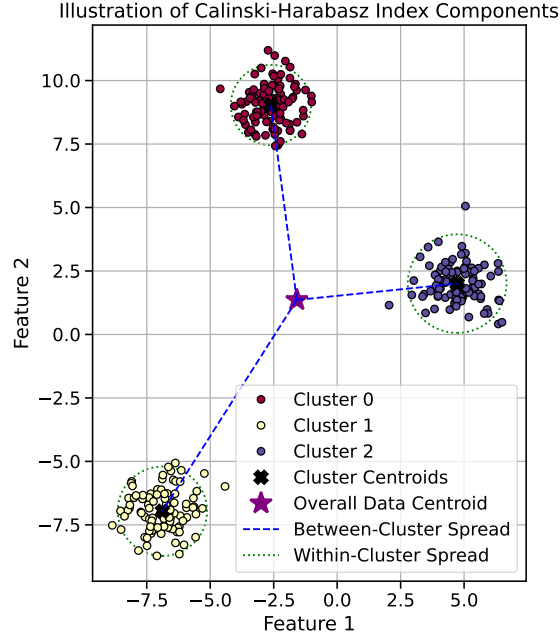


Fig. 4.2 Visualization of between-cluster and within-cluster dispersion components in the Calinski-Harabasz Index framework. The purple star represents the overall data centroid. Blue dashed lines illustrate between-cluster spread (separation) measured from the overall centroid to each individual cluster centroid. Green dotted circles illustrate within-cluster spread (compactness) around each cluster centroid, with radii proportional to the internal point dispersion.

- **Davies-Bouldin Index:** Evaluates the quality of a clustering partition by computing the average similarity between each cluster and its most similar counterpart. Lower DBI values indicate a better partition, characterized by compact and well-separated clusters [151]. The index is formally defined as:

$$DBI = \frac{1}{k} \sum_{i=1}^k \max_{i \neq j} \left(\frac{s_i + s_j}{d_{ij}} \right)$$

where:

- k represents the number of clusters in the clustering solution.
- s_i denotes the average distance between points in cluster C_i and the centroid of cluster C_i , thereby quantifying the scatter or internal dispersion within the cluster.

- d_{ij} denotes the Euclidean distance between the centroids of clusters C_i and C_j , thereby quantifying inter-cluster separation.

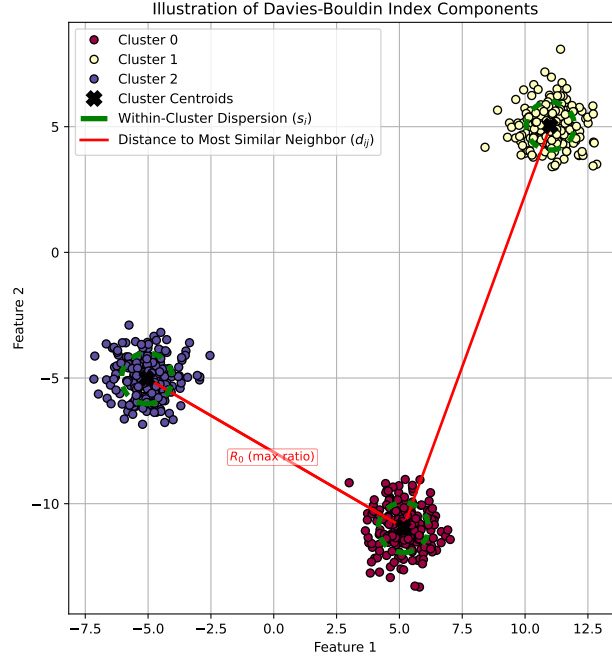


Fig. 4.3 Visualization of cluster compactness and separation in the DBI framework. Green dashed circles around each cluster centroid illustrate within-cluster dispersion (s_i). Red solid lines connect each cluster to its most similar neighbor, with distances (d_{ij}) representing inter-cluster separation. The red line, labeled with largest R_{ij} , highlights the cluster pair with the absolute highest ratio of within-cluster to between-cluster distances, indicating the worst separation in the dataset.

- **Skape Family of Indices:** As discussed in comprehensive detail in the preceding chapter (see subsection 3.6), the Skape (SK) family of indices is specifically designed to emphasize and quantify the disparities and structural differences in samples collected before and after a suspected process event or change point. This objective is achieved by utilizing straightforward yet statistically powerful features extracted from all pairwise Euclidean distances computed between observations in two sequential samples.

These indices, constructed fundamentally from pairwise distance distributions (see Equations 3.4.2 and 3.4.2), centroid separation metrics, and boundary

extreme values, play a crucial and indispensable role in quantifying the structural relationships, geometric configurations, and separation characteristics present within sequential data samples, thereby enabling effective change detection and robust process monitoring.

Such CVIs are essential analytical tools for objectively assessing cluster quality, particularly when clustering techniques are deployed for real-time process monitoring and anomaly detection applications in data-intensive operational environments. By systematically validating clusters through rigorous quantitative measures and statistical criteria, these indices ensure the significance, reliability, and practical relevance of clustering outcomes. This validation process makes certain that any detected deviations, process anomalies, or out-of-control signals are not merely artifacts or false alarms resulting from ineffective clustering algorithms, inappropriate parameter selection, insufficient data quality, or poor model specification.

4.3.4 Monitoring using Cluster Validity

This section establishes the comparative evaluation framework that constitutes a central contribution of this thesis. Building upon the CVI-based monitoring methodology introduced in Chapter 3, where sequential subgroups are treated as evolving clusters and their separability is monitored through Phase I/II control charting, this chapter rigorously evaluates the performance of multiple CVIs under diverse operational scenarios. The primary objective is to demonstrate the superior detection capabilities of the proposed SK family of indices relative to established CVIs including the DI, CHI, SHI, and DBI. While the previous Chapter introduced the SK indices and their theoretical foundation, this chapter provides systematic empirical validation through controlled simulation studies and real-world applications.

Rationale for Comparative Evaluation

The effectiveness of any monitoring framework fundamentally depends on the sensitivity and reliability of its chosen monitoring statistic. In the CVI-based monitoring scheme presented in Chapter 3, the CVI value serves as the primary monitoring statistic analogous to the T^2 statistic in traditional MSPM. Therefore, selecting an appropriate CVI directly determines the framework's ability to detect process anomalies, minimize false alarms, and maintain computational efficiency in high-velocity streaming environments. Existing CVIs were originally designed for static cluster quality assessment in batch clustering contexts, not for sequential monitoring of temporal cluster evolution

in data streams. Consequently, their performance characteristics when adapted to RTM applications remain incompletely understood. The SK indices developed in this thesis were specifically engineered to address limitations observed in traditional CVIs when applied to streaming monitoring contexts, particularly their reduced sensitivity to gradual distributional shifts, vulnerability to high-dimensional sparse data, and inadequate response to correlated process variables.

Evaluation Methodology

Synthetic data generation enables systematic assessment of CVI performance across precisely defined scenarios including:

- Location shifts of varying magnitudes under different correlation structures
- Dispersion changes reflecting process variance alterations
- High-dimensional scenarios examining curse-of-dimensionality effects
- Both Gaussian and non-Gaussian distributional frameworks

For each scenario, the Average Alarm Count (AAC) serves as the primary performance metric, quantifying how rapidly each CVI signals out-of-control conditions following process disturbances. Higher AAC values indicate superior sensitivity and faster detection.

Real-World Classification Tasks

The monitoring framework is applied to labeled datasets where true class boundaries are known, enabling direct validation of anomaly detection accuracy. The Palmer Penguins dataset and mHealth activity recognition dataset provide complementary evaluation contexts, one representing static batch data with clear class separation, the other representing streaming temporal data with complex multivariate dependencies.

Operational Performance Metrics

Beyond detection speed, the evaluation examines computational efficiency (time complexity per monitoring cycle), scalability to high-dimensional feature spaces, and interpretability of out-of-control signals for operational decision-making.

Expected Contributions

Across diverse scenarios, the SK family demonstrates consistently lower AAC values, indicating enhanced sensitivity to process changes compared to traditional CVIs. Different CVIs exhibit varying strengths depending on fault type, dimensionality, and correlation structure. The evaluation provides practitioners with evidence-based selection criteria matching CVI choice to anticipated operational conditions. By demonstrating effective anomaly detection across both synthetic and real-world data, the chapter validates the broader CVI-based monitoring methodology introduced in Chapter 3, confirming its practical utility beyond theoretical development. The subsequent sections present detailed simulation protocols, comprehensive results across all evaluated scenarios, and analysis of findings that inform both theoretical understanding and practical deployment of CVI-based RTM systems.

4.4 Simulation: Cases Evaluated

Due to the considerable computational demands and extended processing time necessary for examining every potential operational scenario, this concentrated on frequently occurring process faults. Specifically, the study addresses mean drift detection and the identification of density-based anomalies in distributed streaming data environments.

To evaluate Phase II detection capabilities, two cluster formations within the streaming dataset are created from distributions sharing identical geometric properties and directional characteristics, but separated by small spatial displacements. These deliberate mean displacements and density modifications serve as controlled representations of realistic process anomalies.

The experimental design specifies $Sim = 10^5$ simulation runs to ensure statistical robustness. Key parameters remain constant throughout the simulation, including dimensional settings (2, 3, 5, 8, 40, and 80) and correlation levels (0.0, 0.3, 0.6, and 0.9) applied consistently during both Phase I baseline establishment and Phase II anomaly detection testing for multivariate Gaussian scenarios. To eliminate potential randomization biases and verify generalizability beyond normal distributions, uniformly distributed observations are additionally generated. These uniform scenarios employ unit spread parameters and centered mean vectors. This dual-distribution approach reveals how the examined metrics respond across different dimensional frameworks and dependency structures.

Maintaining appropriately sized observation groups is essential for achieving optimal balance between computational speed and adequate process variation capture.

Following established statistical process control recommendations, each rational subgroup should include on about 10 observations [135]. This sizing strategy substantially reduces computation time for separability measures while enabling near-instantaneous data processing, meeting the strict temporal requirements of high-speed streaming applications.

4.4.1 Case 1: Drifting the Mean Vector (Shift in Location)

In this scenario, the second cluster experiences a progressive drifting mechanism wherein its mean vector undergoes gradual displacement over successive time intervals (see Figures 4.4 and 4.5). This displacement occurs through systematic incrementation of the mean vector components, causing the second cluster's centroid to progressively migrate through the feature space.

This drifting behavior represents a dynamic transformation in the underlying data-generating process, where cluster characteristics evolve continuously rather than remaining static across temporal iterations. As the centroid position shifts systematically, the entire cluster is transferred through multidimensional space, potentially reflecting gradual trends, environmental changes, or system evolution in the monitored process.

Such progressive drift introduces substantial complications for clustering and monitoring frameworks, as conventional static clustering approaches may fail to accurately maintain cluster boundary definitions or composition integrity over extended periods. To address these challenges, drift detection methodologies, typically grounded in statistical change detection theory, can be deployed to continuously monitor evolving cluster properties, enabling adaptive model updates or recalibration in response to detected drift. This incremental centroid displacement also influences inter-cluster distance metrics and separation quality, potentially generating cluster overlap or revealing emergent distributional patterns.

Properly understanding and modeling this drift phenomenon becomes critical in application domains including anomaly detection systems, sensor network monitoring, and adaptive control systems, where ongoing distributional changes must be identified and managed to maintain reliable operational decisions based on cluster structure analysis. Statistical change detection frameworks combined with clustering methodologies, such as DDM or Page-Hinkley Test (PHT), can be integrated to identify such gradual centroid shifts and trigger model recalibration procedures or operator interventions as required. This integration ensures that the clustering representation remains faithful

to current data characteristics despite ongoing dynamic transformations in underlying feature distributions.

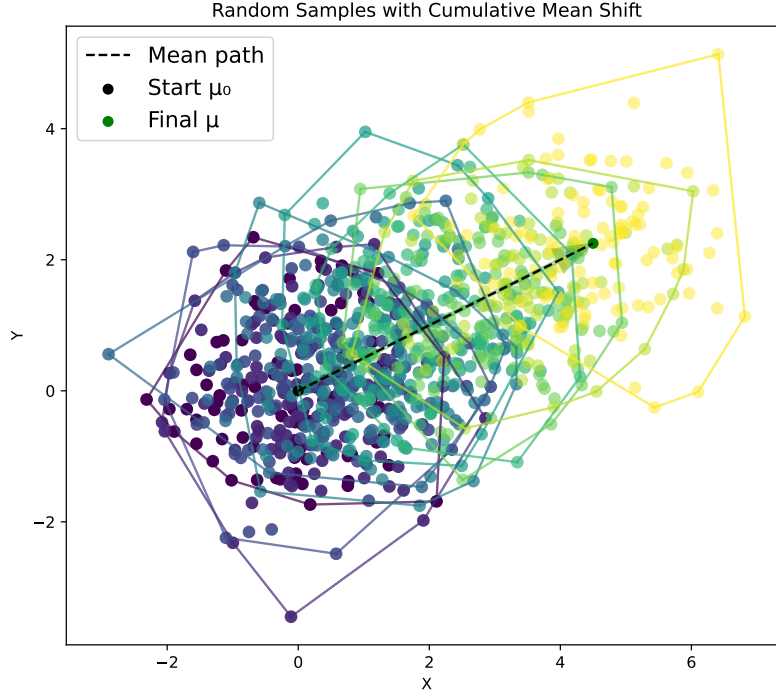


Fig. 4.4 Visualization example of 10 successive randomly sampled clusters with cumulative mean displacement. Each cluster contains 100 observations sampled from a bivariate normal distribution with unit covariance structure. The mean undergoes progressive displacement at each stage by adding an increment proportional to the stage index along a fixed direction vector. Convex hulls delineate the spatial boundary of each cluster, with color and transparency gradually transitioning to reflect temporal progression. The black dashed trajectory line and directional arrow trace the mean displacement path from initial to final stage.

Let μ_1 and μ_2 denote the mean vectors for the first and second clusters, respectively, while Σ_1 and Σ_2 represent their corresponding covariance matrices. In this experimental configuration, the second cluster's centroid undergoes progressive temporal displacement, while its dispersion characteristics (governed by the covariance matrix) remain fixed throughout the simulation.

At the i -th temporal iteration, the second cluster's mean undergoes displacement according to:

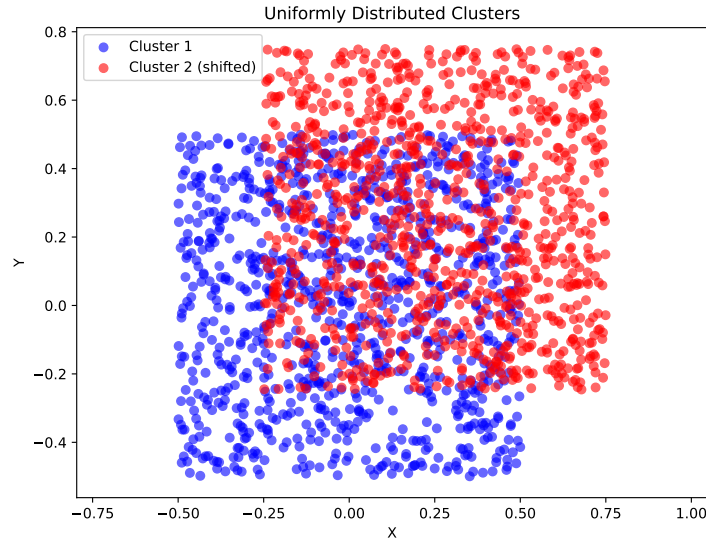


Fig. 4.5 Two uniformly distributed clusters with the reference cluster centered on zero mean vector and the displaced cluster (red) positioned at $x = y = 0.25$. Both clusters exhibit equal spread parameter of 1.

$$\boldsymbol{\mu}_2(i) = \boldsymbol{\mu}_2 + 0.1 \times i$$

Consequently, the centroid of the second cluster advances by 0.1 units along each coordinate axis per iteration. Observations for both clusters are generated using the multivariate Gaussian distribution, with the first cluster's centroid remaining stationary at $\boldsymbol{\mu}_1$, while the second cluster's centroid shifts progressively:

$$\mathbf{X}_2(i) \sim \mathcal{N}(\boldsymbol{\mu}_2(i), \boldsymbol{\Sigma}_2)$$

Here, $\mathbf{X}_2(i)$ denotes the observation set for the second cluster at iteration i , $\boldsymbol{\mu}_2(i)$ represents the varying mean of the second cluster, and $\boldsymbol{\Sigma}_2$ is the fixed covariance matrix governing data dispersion.

The simulation termination criterion is satisfied when all monitoring metrics simultaneously signal out-of-control conditions. To comprehensively capture the temporal response characteristics of each metric across a broader operational context, the cumulative count of all monitoring indicators continues until the final metric registers an out-of-control signal.

Effect on the Data Under this progressive drifting mechanism, the second cluster's centroid moves systematically over time, while data dispersion characteristics (variance and inter-feature correlation) remain constant. The first cluster maintains its stationary position at fixed mean μ_1 , while the second cluster diverges progressively as its centroid advances by 0.1 units per iteration. This drifting process induces temporal changes in the second cluster's spatial location while preserving its variance structure unchanged.

4.4.2 Case 2: Scaling the Data Points (Change in Spread)

In this scenario, the second cluster undergoes a systematic scaling transformation. Here, observations belonging to the second cluster are initially generated using fixed centroid and covariance parameters, after which a progressively increasing scaling factor is applied to these observations (see Figures 4.6 and 4.7). This factor grows systematically over temporal iterations, causing the second cluster to exhibit progressively increasing dispersion.

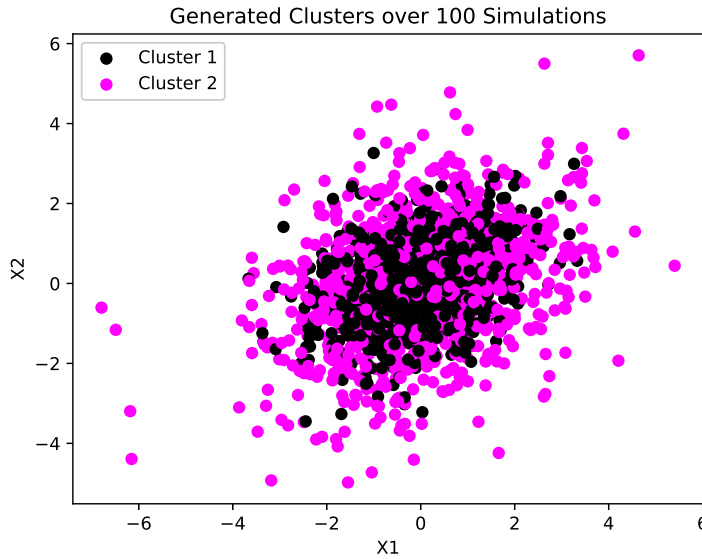


Fig. 4.6 Progressive increase in data dispersion from initial sample (black dots) to modified sample (magenta dots), both generated from multivariate normal distribution. The impact of the scaling factor demonstrates the magnitude of dispersion increase after approximately 100 simulation iterations.

Let μ_1 and μ_2 denote the mean vectors for the first and second clusters, respectively, while Σ_1 and Σ_2 represent their corresponding covariance matrices. In this experimental configuration, observations for the second cluster are generated using fixed centroid μ_2

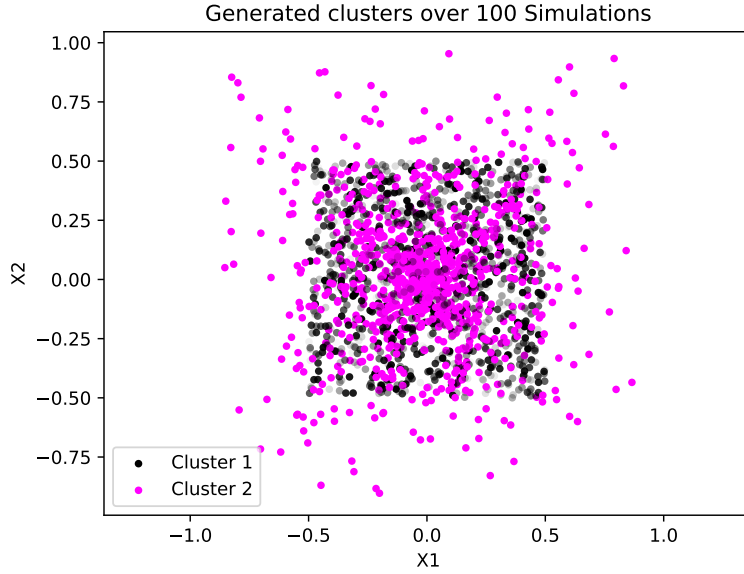


Fig. 4.7 Progressive increase in data dispersion from initial sample (black dots) to modified sample (magenta dots), generated from multivariate uniform distribution. The impact of the scaling factor demonstrates the magnitude of dispersion increase after approximately 100 simulation iterations.

and covariance structure Σ_2 . However, following initial generation, a multiplicative scaling factor $\delta(i)$ is applied to systematically increase the dispersion of the second cluster.

Initially, the scaling factor is initialized to unity, and subsequently incremented by 0.01 with each temporal iteration. The scaling factor evolves according to:

$$\delta(i) = 1 + 0.01 \times (i + 1)$$

After generating the second cluster's observation set $\mathbf{X}_2(i)$, these points undergo multiplicative scaling by factor $\delta(i)$:

$$\mathbf{X}'_2(i) = \mathbf{X}_2(i) \times \delta(i)$$

Here, $\mathbf{X}_2(i)$ represents the original observation set for the second cluster, generated using centroid μ_2 and covariance Σ_2 , while $\mathbf{X}'_2(i)$ denotes the scaled observation set, which exhibits progressively increasing dispersion over temporal iterations as $\delta(i)$ grows.

Following the simulation procedure conducted at moderate displacement magnitudes, an additional experimental procedure is implemented to evaluate metric responsiveness under density modification scenarios. Due to limited detection sensitivity observed for

most monitoring indicators under this condition, the termination criterion is modified. In this revised procedure, the simulation terminates when the first monitoring metric or metrics register in-control signals, representing a more stringent evaluation of early detection capability.

Effect on the Data Under this scaling transformation, observations for the second cluster are generated following standard procedures, but their spatial dispersion is systematically amplified through application of the scaling factor. As the scaling factor increases with successive iterations, the second cluster’s observations become progressively more dispersed throughout the feature space. The first cluster maintains unchanging characteristics in terms of both centroid location and dispersion, while the second cluster exhibits increasing spread as the simulation progresses. This scaling process systematically increases the dispersion (variance) of the second cluster’s observations over temporal iterations, while the centroid location (mean) remains fixed at its initial position.

4.4.3 Simulation Results

The following tables present the AAC recorded across the evaluated experimental scenarios.

4.4.4 Simulation Conclusions

Location shift — Multivariate Normal. Across most combinations of dimension d and correlation ρ , SK indices (especially SK_{std} and often SK_{max}) attain the highest AAC, indicating superior sensitivity to mean shifts under Gaussian conditions (Table 4.1). At smaller to moderate dimensions ($d = 2, 3, 5$), SK consistently leads or ties with DBI and CHI. As d increases ($d \geq 8$), CHI and DBI occasionally attain comparable or slightly higher AAC values, particularly under weak correlation ($\rho = 0, 0.3$). This suggests a gradual convergence of performance at higher dimensionalities where the benefit of SK’s cumulative separation diminishes. As correlation strengthens ($\rho \geq 0.6$), SK indices generally regain dominance, benefiting from anisotropic structure that enhances their aggregation-based sensitivity. The DI and SHI indices remain conservative and comparatively less responsive to mean shifts throughout.

Spread change — Multivariate Normal. For dispersion alterations, SK metrics show the strongest performance across nearly all reported d and ρ values (Table 4.2).

d	ρ	SHI	DI	CHI	SK_{max}	SK_{std}	DBI	$Sims$	μ_{out}
2	0.0	0.54	0.248	0.587	0.577	0.599	0.586	3.954	0.04
	0.3	0.524	0.24	0.59	0.604	0.616	0.587	3.972	0.04
	0.6	0.523	0.245	0.585	0.584	0.598	0.586	4.106	0.041
	0.9	0.506	0.246	0.566	0.573	0.589	0.563	4.188	0.042
3	0.0	0.57	0.256	0.604	0.606	0.609	0.602	3.767	0.038
	0.3	0.547	0.232	0.617	0.628	0.64	0.615	3.871	0.039
	0.6	0.545	0.241	0.602	0.602	0.614	0.603	4.052	0.041
	0.9	0.52	0.248	0.57	0.588	0.599	0.569	4.231	0.042
5	0.0	0.608	0.251	0.635	0.635	0.625	0.633	3.528	0.035
	0.3	0.578	0.226	0.642	0.647	0.66	0.641	3.705	0.037
	0.6	0.574	0.235	0.615	0.621	0.629	0.618	4.048	0.04
	0.9	0.532	0.241	0.587	0.598	0.61	0.589	4.256	0.043
8	0.0	0.642	0.239	0.662	0.658	0.635	0.66	3.31	0.033
	0.3	0.617	0.205	0.675	0.673	0.675	0.675	3.582	0.036
	0.6	0.595	0.223	0.632	0.636	0.643	0.637	3.97	0.04
	0.9	0.541	0.237	0.585	0.605	0.608	0.589	4.249	0.042
40	0.0	0.725	0.219	0.731	0.728	0.626	0.732	2.583	0.026
	0.3	0.681	0.189	0.702	0.698	0.695	0.705	3.406	0.034
	0.6	0.623	0.206	0.65	0.66	0.664	0.656	3.984	0.04
	0.9	0.554	0.225	0.6	0.622	0.625	0.603	4.269	0.043
80	0.0	0.756	0.206	0.76	0.763	0.601	0.76	2.33	0.023
	0.3	0.693	0.181	0.715	0.704	0.697	0.716	3.403	0.034
	0.6	0.621	0.209	0.65	0.664	0.664	0.654	3.951	0.04
	0.9	0.553	0.218	0.6	0.628	0.634	0.603	4.29	0.043

Table 4.1 AAC in Location Shift (Multivariate Normal Distribution)

SK_{max} consistently achieves, or ties for, the highest AAC across conditions. There are no cases where DI, CHI, SHI, or DBI surpass SK; their AACs remain stable but lower, confirming SK's superior responsiveness to scale changes in Gaussian data. The uniform domination of SK underscores its ability to capture aggregate dispersion changes efficiently.

Location shift — Multivariate Uniform. Under uniform sampling, SK_{max} attains the highest AAC at every reported dimension (Table 4.3). The monotonic increase of SK scores with rising d corresponds to improved detection strength as higher-dimensional separation effects accumulate. All other indices (SHI, CHI, DI, DBI) consistently trail behind SK. This pattern indicates that mean-shift sensitivity remains robust in non-Gaussian, bounded-support environments.

d	ρ	SHI	DI	CHI	SK_{max}	SK_{std}	DBI	$Sims$	sd_{out}
2	0.0	0.372	0.354	0.358	0.433	0.444	0.363	7.187	0.072
	0.3	0.375	0.357	0.36	0.433	0.433	0.361	7.138	0.071
	0.6	0.375	0.363	0.365	0.425	0.424	0.366	7.232	0.072
	0.9	0.369	0.361	0.365	0.426	0.425	0.362	7.093	0.071
3	0.0	0.375	0.364	0.353	0.449	0.45	0.354	7.196	0.072
	0.3	0.379	0.361	0.364	0.443	0.442	0.365	7.311	0.073
	0.6	0.384	0.372	0.37	0.427	0.426	0.372	7.303	0.073
	0.9	0.373	0.368	0.361	0.428	0.426	0.359	7.256	0.073
5	0.0	0.381	0.351	0.355	0.472	0.47	0.358	7.116	0.071
	0.3	0.385	0.368	0.365	0.445	0.444	0.366	7.344	0.073
	0.6	0.386	0.385	0.365	0.437	0.432	0.365	7.56	0.076
	0.9	0.376	0.372	0.372	0.43	0.431	0.37	7.355	0.074
8	0.0	0.388	0.332	0.357	0.492	0.487	0.359	7.055	0.071
	0.3	0.414	0.347	0.389	0.469	0.456	0.39	7.332	0.073
	0.6	0.409	0.368	0.388	0.455	0.452	0.39	7.394	0.074
	0.9	0.386	0.373	0.372	0.438	0.43	0.372	7.389	0.074
40	0.0	0.375	0.263	0.319	0.554	0.517	0.324	6.29	0.063
	0.3	0.423	0.365	0.381	0.483	0.472	0.383	7.577	0.076
	0.6	0.406	0.38	0.383	0.465	0.46	0.386	7.773	0.078
	0.9	0.392	0.374	0.378	0.444	0.441	0.379	7.489	0.075
80	0.0	0.357	0.217	0.294	0.595	0.511	0.298	5.934	0.059
	0.3	0.427	0.363	0.39	0.489	0.474	0.39	7.716	0.077
	0.6	0.409	0.385	0.386	0.467	0.459	0.387	7.669	0.077

Table 4.2 AAC in Spread Change (Multivariate Normal Distribution)

d	SHI	DI	CHI	SK_{max}	SK_{std}	DBI	$Sims$	μ_{out}
2	0.403	0.305	0.479	0.706	0.675	0.462	3.46	0.035
3	0.442	0.306	0.517	0.706	0.68	0.504	3.298	0.033
5	0.493	0.303	0.553	0.703	0.682	0.543	3.084	0.031
8	0.542	0.303	0.589	0.715	0.664	0.584	2.885	0.029
40	0.685	0.316	0.697	0.773	0.598	0.696	2.258	0.023
80	0.736	0.331	0.741	0.806	0.54	0.741	2.039	0.02

Table 4.3 AAC in Location Shift (Multivariate Uniform Distribution)

d	SHI	DI	CHI	SK_{max}	SK_{std}	DBI	$Sims$	sd_{out}
2	0.377	0.36	0.371	0.478	0.473	0.371	7.316	0.073
3	0.372	0.359	0.369	0.486	0.476	0.371	7.326	0.073
5	0.364	0.351	0.355	0.489	0.482	0.355	7.133	0.071
8	0.361	0.338	0.353	0.5	0.478	0.354	6.986	0.07
40	0.331	0.261	0.311	0.556	0.476	0.314	6.239	0.062
80	0.312	0.217	0.288	0.602	0.452	0.292	5.873	0.059

Table 4.4 AAC in Spread Change (Multivariate Uniform Distribution)

Spread change — Multivariate Uniform. For spread changes under uniform sampling (Table 4.4), SK_{max} again yields the highest AAC across all dimensions. The margin of superiority grows with d , showing improved relative performance in higher-dimensional space. No competing index surpasses SK in any condition, confirming its generality as a scale-sensitive metric beyond the normal distribution framework.

Synthesis and rationale. The AAC patterns exhibit a coherent structure across all experiments. Indices that aggregate weak directional evidence and respond to tail or spread variations (such as SK_{std} and SK_{max}) consistently outperform compactness-based measures. Under normal models with mean shifts, SK indices exhibit dominant or near-dominant AACs across most (d, ρ) , with high-dimensional low-correlation conditions being the only partial exceptions where CHI or DBI can occasionally tie or marginally exceed SK. For spread-change detection, SK performance is unequivocally superior in both normal and uniform settings. These outcomes suggest that SK indices maintain broad robustness across distribution families and parameter regimes, leveraging their sensitivity to cumulative separation and dispersion even when tail structure is attenuated.

4.5 Real-World Data

Combining simulation studies with authentic operational data represents a fundamental requirement for establishing methodological credibility, contextual validity, and industrial relevance of proposed monitoring techniques. While computational simulations offer controlled experimental conditions that enable systematic hypothesis testing and parameter exploration, they necessarily operate under idealized assumptions that may

inadequately represent the complexity inherent in actual production environments. Authentic operational data provides essential context for validating simulation-derived models against real-world conditions, exposing inconsistencies, unexpected phenomena, or confounding factors absent from simplified simulation frameworks. This synergistic integration substantially strengthens both the internal reliability of experimental findings and the external applicability of predictive models to genuine industrial applications, effectively bridging theoretical constructs with operational realities.

Identifying meaningful process variations within operational datasets requires systematic methods for detecting structural differences or temporal evolutions that indicate substantive changes in system behavior. Supervised classification provides a statistically rigorous and interpretable framework for this analytical task by mapping data observations to predefined category labels, thereby revealing underlying structural patterns and establishing verifiable ground truth for performance evaluation. By constructing classification experiments that integrate the previously described MSPM frameworks with clustering validity assessment protocols, researchers can rigorously quantify monitoring metric performance against verified class labels or expert annotations. Both static batch datasets and dynamic streaming datasets are utilized throughout evaluation procedures, supplying diverse experimental conditions that replicate operational challenges including temporal distribution evolution, concept drift, high-dimensional feature spaces, and intricate variable dependencies.

This methodological strategy enables precise quantitative evaluation of detection metrics while simultaneously enhancing interpretability of anomaly identification by harmonizing statistical monitoring approaches with supervised learning validation. Incorporating classification tasks within experimental design consequently creates a robust linkage between theoretical model assessment and practical operational decision-making, establishing a validated pathway for refining monitoring systems suitable for complex, dynamic industrial environments.

4.5.1 The Palmer Penguins Dataset (Batch Data)

The Palmer Penguins dataset represents an important educational and scientific resource that documents morphometric measurements from three distinct penguin species: Adélie (*Pygoscelis adeliae*), Chinstrap (*Pygoscelis antarcticus*), and Gentoo (*Pygoscelis papua*) (refer to Figure 4.8), all of which breed across islands within the Palmer Archipelago situated near Antarctica. Originally assembled through a comprehensive long-term ecological research initiative [152], this collection encompasses biological measurements covering bill length and depth, flipper length, body mass, sex determination, and

geographical location data from 344 individual specimens. Later curated and made accessible through a dedicated R package designed for educational applications [153], the dataset has achieved widespread recognition as a valuable tool for statistical analysis, supervised classification tasks, and data visualization demonstrations.

These three species demonstrate distinctive biological traits and behavioral adaptations that enrich ecological investigations. Adélie penguins are relatively diminutive in stature, feature a characteristic white periocular ring, and display affinity for ice-dominated habitats. Chinstrap penguins, identifiable by the distinctive narrow black marking extending across their lower mandible, demonstrate strong swimming abilities and exhibit particular skill in establishing nests on steep rocky terrain. Gentoo penguins, representing the largest of these three species, are distinguished by their striking orange-colored bills and exceptional aquatic velocity, showing preference for less ice-covered, relatively temperate breeding locations. These interspecific morphological and behavioral variations reflect evolutionary adaptations to distinct ecological niches and exemplify morphological diversification in response to environmental pressures.

The dataset's origins in rigorous field research underscore the critical importance of incorporating authentic observational data into experimental methodologies, particularly where computational simulations may inadequately capture the inherent complexity of biological systems. Although simulations provide standardized, reproducible experimental environments, they inherently depend on simplifying assumptions. Empirical observations obtained from field studies, such as those conducted under the Palmer Long-Term Ecological Research (LTER) program, ground analytical models in empirical evidence [152], enabling both researchers and students to confront practical challenges including incomplete data records, natural biological variation, and species-level discrimination. Within this comprehensive framework, the Palmer Penguins dataset [153] functions not merely as a vehicle for developing technical analytical skills but also as a catalyst for fostering ecological awareness and cultivating appreciation for biodiversity conservation.

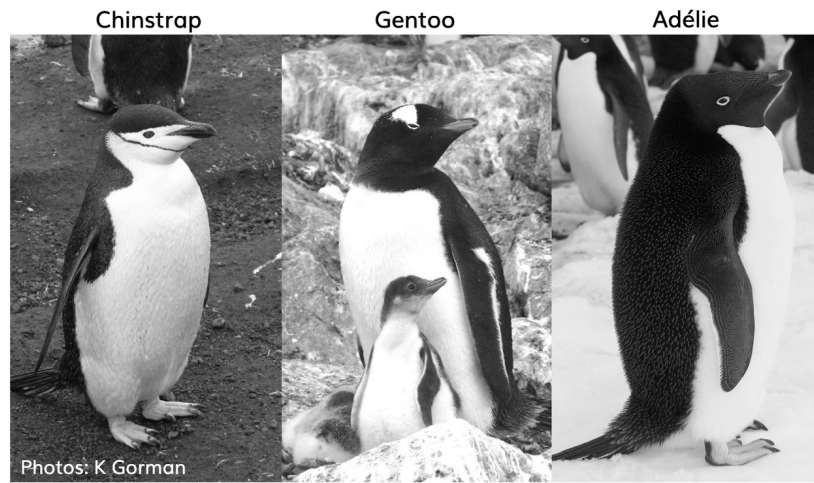


Fig. 4.8 A photo taken by Dr. Kristen Gorman illustrating the three species of penguins in the dataset. Adélie (*Pygoscelis adeliae*), Chinstrap (*Pygoscelis antarcticus*), and Gentoo (*Pygoscelis papua*)

Dataset Characteristics

A comprehensive analytical treatment may surpass the scope of this investigation; nevertheless, visual examination of multivariate data (see Figure 4.9) remains essential and serves as a foundational preliminary stage in investigating morphological and ecological distinctions across species. Prior to implementing formal statistical methodologies, graphical exploration can reveal latent structural patterns, variable relationships, and species-specific clustering characteristics. Within the Palmer Penguins dataset, quantitative morphometric attributes including bill dimensions, flipper measurements, and body mass, which have been influenced by evolutionary pressures and environmental factors, provide meaningful perspectives on interspecific variation among Adélie, Chinstrap, and Gentoo penguin populations.

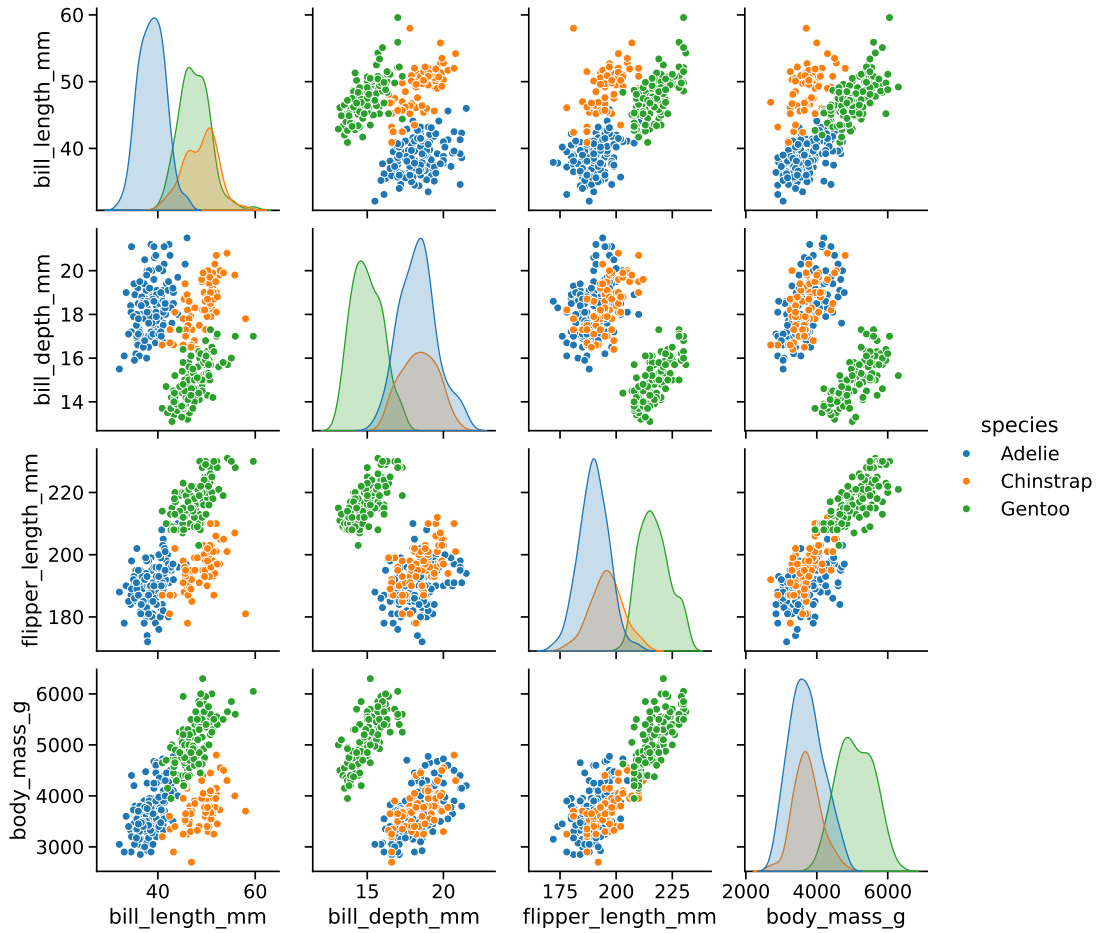


Fig. 4.9 Intuitive look at how bill length, bill depth, flipper size, and body mass differ across species.

Results

The findings from various clustering validity measures applied to quantify species distinctiveness in the Palmer Penguin dataset are presented in Table 4.5, featuring pairwise comparisons among Adelie (A), Chinstrap (C), and Gentoo (G) species. The assessment incorporates the SHI, DBI, DI, CHI, SK_{max} , SK_{std} , alongside n_{sim} (denoting simulation replication count).

The pairwise evaluation in Table 4.5 indicates that separability is strongest for comparisons involving Gentoo. Specifically, SK_{max} attains 0.981 for C–G (with $SK_{std} = 0.952$) and remains very high for A–G and G–A (0.958/0.937 for SK_{max} ; 0.931/0.897 for SK_{std}), evidencing pronounced interspecific distinction whenever Gentoo participates (Table 4.5). CHI aligns with this pattern, with larger values for A–G/G–A and C–G/G–C (e.g., 0.729/0.690 and 0.814/0.786), while DI is comparatively small for A–C (0.065)

and larger for A–G/G–A and C–G/G–C (0.636–0.728), indicating greater separation in the latter pairs (Table 4.5). DBI takes higher values for A–C and for the first C–G pair (0.031–0.040) and zeros for the remaining Gentoo-involving rows, therefore indicating strongest separability for A–C and that C–G entry and weakest for the zero-valued Gentoo pairs. SHI is small for A–C and the first C–G row (0.057–0.066) but substantially larger for Gentoo-involving pairs later in the table (0.667–0.799), again reinforcing the same ordering of distinctiveness (Table 4.5). The simulation replication counts n_{sim} vary across pairs (e.g., 2,278; 7,021; 10,585); the table records this variability without attributing a cause (Table 4.5). Overall, the indices consistently rank C–G/G–C as most distinct, A–G/G–A as strongly distinct, and A–C as least distinct in the configurations reported, an ordering that is coherent across SK_{max} , SK_{std} , CHI, DI, DBI, and SHI (Table 4.5).

x	y	SHI	DBI	DI	CHI	SK_{max}	SK_{std}	n_{sim}
A	C	0.057	0.031	0.065	0.053	0.030	0.028	2278
C	A	0.066	0.040	0.064	0.062	0.077	0.076	10585
A	G	0.701	0.000	0.650	0.729	0.958	0.931	7021
G	A	0.667	0.000	0.636	0.690	0.937	0.897	10585
C	G	0.799	0.000	0.720	0.814	0.981	0.952	7021
G	C	0.789	0.000	0.728	0.786	0.946	0.874	2278

Table 4.5 Penguin Classification Result (where, Class A is Adelie, C is Chinstrap, and G is Gentoo).

4.5.2 mHealth Dataset (Streaming Data)

The mHealth (Mobile Health) dataset [154] encompasses multimodal recordings of body motion parameters and physiological vital signs collected from ten volunteer participants exhibiting diverse demographic characteristics, while performing various physical activities. Data acquisition was executed using Shimmer2 wearable sensor devices strategically positioned at three anatomical locations: the chest, right wrist, and left ankle. These sensor units were employed to capture kinematic parameters, including triaxial acceleration, angular velocity, and magnetic field orientation vectors, associated with distinct body segments during activity execution. Additionally, the sensor unit positioned on the chest acquired 2-lead electrocardiogram (ECG) signals, which, although not utilized in the primary activity classification model, remain avail-

able for subsequent investigations of cardiac health status, arrhythmia detection, or the physiological effects of exercise on cardiovascular function parameters. The dataset encompasses recordings from 12 distinct activities, captured using 3 sensor devices across 10 volunteer subjects. This collection comprises temporally synchronized recordings of both kinematic body motion and physiological vital sign data obtained from ten participants executing twelve different physical activities. Data acquisition utilized Shimmer2 wearable sensing technology, with individual sensor units positioned on the subject's chest, right wrist, and left ankle, fastened using elastic retention straps. The strategic placement of these sensor modules enabled comprehensive measurement of diverse kinematic parameters across multiple body segments, including linear acceleration, rotational rate, and magnetic field vector orientation, thereby providing a holistic representation of body dynamics during physical activity execution. Beyond kinematic data, the thoracic sensor unit acquired 2-lead electrocardiographic (ECG) recordings. Although these ECG measurements were not directly incorporated into the primary activity classification model, they remain available for prospective investigations, including fundamental cardiac monitoring, cardiac arrhythmia identification, or examination of exercise-induced alterations in electrocardiographic characteristics. All sensor modalities were sampled at a uniform rate of 50 Hz, considered sufficient for capturing human activity dynamics. To ensure precise activity annotation and enable subsequent validation, each recording session was simultaneously documented via video capture. Data collection occurred in naturalistic, non-laboratory environments, with participants encouraged to execute activities naturally, receiving only general instructions to perform each activity optimally. This diverse and minimally constrained experimental protocol enhances dataset generalizability to authentic daily activities, accounting for variations in engaged body segments, movement intensity levels (e.g., cycling versus sitting), and activity speed and dynamism (e.g., running versus standing motionless).

The activities (see Table 4.6), characterized by heterogeneous durations and intensity profiles, furnish a robust dataset for activity recognition model development and extended investigations in human motion analysis. The diversity of movements, spanning from low-intensity tasks such as sitting and supine positioning, to high-intensity exercises such as running and cycling, ensures comprehensive representation of daily physical behaviors.

Table 4.6 Activity set with durations and repetition counts

Label	Activity	Type	Duration / Repetitions
L1	Standing motionless	Low intensity	1 minute
L2	Sitting and relaxing	Low intensity	1 minute
L3	Supine positioning	Low intensity	1 minute
L4	Walking	Moderate intensity	1 minute
L5	Stair ascent	Moderate intensity	1 minute
L6	Forward waist flexion	Repetitive exercise	20 repetitions
L7	Frontal arm elevation	Repetitive exercise	20 repetitions
L8	Knee flexion (crouching)	Repetitive exercise	20 repetitions
L9	Cycling	High intensity	1 minute
L10	Jogging	High intensity	1 minute
L11	Running	High intensity	1 minute
L12	Sagittal jumping	Repetitive exercise	20 repetitions

Individual subject data are stored in separate log files¹ designated, where each file contains sensor measurements recorded across multiple sensing modalities during activity monitoring. Data organization follows a row-sample, column-variable structure, with each row representing a temporal sample and each column (V1-V23) representing distinct sensor measurements. Log files document multiple sensor reading types, with individual columns dedicated to specific measurements from sensors positioned on different anatomical locations. The initial three columns capture triaxial acceleration data (m/s^2) from the thoracic sensor along X, Y, and Z axes.

Columns four and five correspond to electrocardiogram (ECG) signals from leads 1 and 2, respectively, quantified in millivolts (mV). Subsequent columns include triaxial acceleration data from a left-ankle sensor, followed by triaxial gyroscope measurements from the identical sensor, and triaxial magnetometer readings from the left ankle. Additional columns capture analogous sensor data from a right-lower-arm sensor, encompassing acceleration, gyroscope, and magnetometer measurements. The terminal column contains activity labels, where a value of “0” denotes the null class (absence of activity). Activity labels, such as “4” for walking, classify data according to the executed activity type. This multi-sensor configuration enables comprehensive physical activity analysis by providing detailed measurements of movement kinematics, spatial

¹`mHealth_subjectSUBJECT_ID.log`

orientation, and vital physiological signals. Inter-variable correlation structure is illustrated in Figure 4.10.

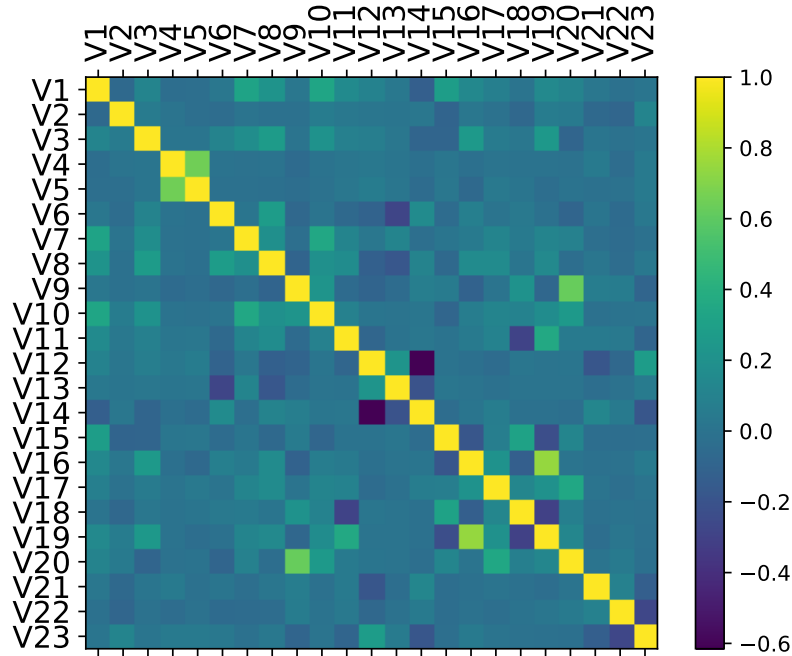


Fig. 4.10 Correlation matrix of mHealth Dataset

Results

Obtaining reliable outcomes, particularly when evaluating clustering performance and validity metrics, critically depends on the sample size utilized in experimental procedures. Specifically, reducing sample size frequently leads to deterioration in the quality and accuracy of performance measures. This reduction renders clustering results less stable, as metrics become increasingly susceptible to noise and variability when operating with limited datasets. Consequently, to preserve the validity and reliability of clustering evaluations, maintaining adequate sample size remains essential. However, for computational efficiency considerations, sample size should be minimized to the extent possible without compromising the integrity of performance evaluation. Therefore, achieving appropriate balance becomes vital to avoid detrimental effects on overall cluster validity assessment and metric reliability. Given the velocity requirements of data streams, which necessitate small batch sizes to attain higher processing throughput, and considering the range of observations employed as batch sizes across different investigations [155], spanning from $2 \leq n \leq 128$, the selected batch size was established

at 3 observations. Table 4.7 summarizes the discriminative effectiveness of various CVIs, including the SHI, DI, CHI, SK_{max} , SK_{std} , and DBI, in differentiating between activity class pairs from the mHealth dataset. Critical examination reveals insights regarding both the discriminative capacity of these indices across movement categories and the practical utility of CVIs metrics for human activity recognition research.

All values in Table 4.7 are interpreted on $[0, 1]$ with larger scores indicating better discriminative performance for each index. The batch size was set to three observations to balance streaming throughput with acceptable stability of the reported CVIs across pairwise directions.

Sedentary pairs (L1–L3) exhibit near-ceiling separability for SHI, DI, CHI, and SK_{max} in both orderings of (1,2) and (2,3), and in at least one ordering for (1,3), reflecting clean posture distinctions. Directional asymmetry appears for several pairs (i, j) versus (j, i) , consistent with small-sample variability in directional estimates. For static–dynamic contrasts, SK_{max} typically remains very high, while SHI, DI, and CHI decrease relative to the sedentary baseline, indicating greater overlap or dispersion under movement. For dynamic–dynamic pairs with similar kinematics (for example, walking versus stairs, or jogging versus running/jumping), overlap-sensitive indices are smallest, although SK_{max} often identifies a strong separating direction in at least one ordering.

Two index-level regularities are evident. First, SK_{max} frequently saturates near unity across static–static, static–dynamic, and several dynamic–dynamic contrasts, whereas SK_{std} is lower and more variable, indicating that a single dominant direction can separate classes even when separation stability and compactness–separation ratios are conservative. Second, DBI shows weak performance across many pairs in the table, with values concentrated near zero and offering limited discrimination relative to the other indices; consequently, DBI contributes less to ranking activity-pair separability in this setting.

Table 4.7 Metric Performance for mHealth Dataset (subject1)

$\text{Pair}^{(i,j)}$	$\text{SHI}^{(i,j),(j,i)}$		$\text{DI}^{(i,j),(j,i)}$		$\text{CHI}^{(i,j),(j,i)}$		$\text{SK}_{max}^{(i,j),(j,i)}$		$\text{SK}_{std}^{(i,j),(j,i)}$		$\text{DBI}^{(i,j),(j,i)}$	
(1, 2)	0.999	0.999	0.968	0.965	0.984	0.984	1.000	1.000	0.023	0.164	0.000	0.000
(1, 3)	1.000	1.000	1.000	0.998	1.000	1.000	1.000	1.000	0.023	0.621	0.000	0.000
(1, 4)	0.606	0.215	0.280	0.116	0.376	0.131	0.999	0.001	0.958	0.000	0.062	0.005
(1, 5)	0.516	0.336	0.104	0.086	0.140	0.112	0.999	0.011	0.894	0.022	0.023	0.009
(1, 6)	0.533	0.191	0.179	0.102	0.231	0.105	0.992	0.002	0.759	0.000	0.038	0.006
(1, 7)	0.433	0.182	0.249	0.125	0.304	0.137	0.992	0.000	0.870	0.004	0.145	0.007
(1, 8)	0.532	0.225	0.209	0.105	0.246	0.137	0.984	0.001	0.702	0.005	0.031	0.001
(1, 9)	0.610	0.261	0.279	0.121	0.314	0.154	1.000	0.000	0.781	0.001	0.018	0.000
(1, 10)	0.655	0.178	0.135	0.084	0.177	0.092	1.000	0.001	0.989	0.002	0.035	0.012
(1, 11)	0.640	0.076	0.180	0.187	0.218	0.131	1.000	0.000	0.998	0.000	0.077	0.016
(1, 12)	0.746	0.310	0.122	0.093	0.166	0.120	1.000	0.005	0.972	0.016	0.043	0.027
(2, 3)	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.081	0.498	0.000	0.000
(2, 4)	0.673	0.228	0.265	0.131	0.398	0.130	1.000	0.001	0.995	0.000	0.069	0.001
(2, 5)	0.795	0.358	0.107	0.105	0.166	0.113	1.000	0.013	0.987	0.022	0.013	0.002
(2, 6)	0.738	0.288	0.183	0.133	0.272	0.153	1.000	0.003	0.976	0.006	0.018	0.001
(2, 7)	0.549	0.182	0.219	0.126	0.292	0.132	1.000	0.000	0.966	0.000	0.185	0.025
(2, 8)	0.818	0.309	0.233	0.148	0.306	0.179	1.000	0.001	0.960	0.005	0.004	0.000
(2, 9)	0.700	0.268	0.241	0.101	0.314	0.138	1.000	0.000	0.969	0.001	0.029	0.000
(2, 10)	0.736	0.191	0.127	0.100	0.184	0.109	1.000	0.001	0.999	0.001	0.049	0.008
(2, 11)	0.709	0.180	0.157	0.116	0.207	0.125	1.000	0.000	1.000	0.000	0.098	0.019
(2, 12)	0.861	0.346	0.101	0.092	0.168	0.109	1.000	0.001	0.996	0.014	0.052	0.014
(3, 4)	0.784	0.244	0.267	0.141	0.426	0.143	1.000	0.003	0.999	0.000	0.035	0.000
(3, 5)	0.994	0.511	0.157	0.148	0.270	0.154	1.000	0.011	0.997	0.018	0.002	0.000
(3, 6)	0.989	0.541	0.362	0.275	0.511	0.267	1.000	0.003	0.994	0.004	0.000	0.000
(3, 7)	0.785	0.213	0.256	0.141	0.366	0.152	1.000	0.000	0.996	0.000	0.093	0.000
(3, 8)	1.000	0.578	0.412	0.289	0.560	0.330	1.000	0.003	0.990	0.006	0.000	0.000
(3, 9)	0.997	0.433	0.361	0.178	0.458	0.233	1.000	0.000	0.993	0.000	0.000	0.000
(3, 10)	0.834	0.192	0.116	0.106	0.180	0.118	1.000	0.001	1.000	0.002	0.042	0.013
(3, 11)	0.774	0.164	0.143	0.106	0.218	0.112	1.000	0.000	1.000	0.000	0.091	0.015
(3, 12)	0.951	0.343	0.097	0.070	0.195	0.092	1.000	0.006	1.000	0.021	0.032	0.011
(4, 5)	0.042	0.130	0.049	0.123	0.042	0.134	0.010	0.195	0.004	0.187	0.013	0.074
(4, 6)	0.075	0.150	0.098	0.144	0.072	0.141	0.006	0.497	0.000	0.349	0.005	0.013
(4, 7)	0.081	0.041	0.091	0.051	0.068	0.039	0.005	0.067	0.000	0.049	0.004	0.002
(4, 8)	0.083	0.140	0.094	0.121	0.084	0.138	0.005	0.704	0.000	0.638	0.010	0.028
(4, 9)	0.076	0.103	0.098	0.092	0.078	0.105	0.013	0.544	0.001	0.519	0.006	0.035
(4, 10)	0.040	0.073	0.041	0.066	0.034	0.069	0.333	0.005	0.254	0.005	0.011	0.029
(4, 11)	0.071	0.116	0.067	0.103	0.059	0.105	0.687	0.000	0.498	0.000	0.011	0.028
(4, 12)	0.026	0.080	0.027	0.068	0.024	0.069	0.108	0.036	0.079	0.024	0.009	0.071
(5, 6)	0.075	0.035	0.071	0.033	0.065	0.028	0.012	0.190	0.012	0.147	0.021	0.010
(5, 7)	0.165	0.021	0.145	0.035	0.155	0.020	0.041	0.010	0.017	0.008	0.039	0.001
(5, 8)	0.077	0.035	0.072	0.031	0.073	0.032	0.018	0.348	0.023	0.327	0.033	0.011
(5, 9)	0.065	0.032	0.076	0.031	0.068	0.031	0.021	0.241	0.022	0.229	0.029	0.015
(5, 10)	0.138	0.079	0.100	0.066	0.121	0.076	0.734	0.004	0.576	0.007	0.035	0.019
(5, 11)	0.222	0.129	0.157	0.105	0.179	0.115	0.930	0.000	0.753	0.000	0.056	0.031
(5, 12)	0.103	0.081	0.091	0.066	0.090	0.070	0.373	0.012	0.286	0.011	0.027	0.031
(6, 7)	0.129	0.049	0.119	0.065	0.120	0.051	0.281	0.000	0.226	0.001	0.043	0.002
(6, 8)	0.057	0.045	0.071	0.045	0.049	0.045	0.011	0.065	0.012	0.067	0.021	0.033
(6, 9)	0.043	0.042	0.075	0.058	0.040	0.038	0.010	0.014	0.004	0.011	0.013	0.011
(6, 10)	0.120	0.114	0.100	0.095	0.099	0.095	0.933	0.002	0.795	0.003	0.017	0.018
(6, 11)	0.185	0.163	0.124	0.120	0.140	0.131	0.993	0.000	0.901	0.000	0.026	0.023
(6, 12)	0.059	0.131	0.051	0.072	0.046	0.079	0.703	0.003	0.537	0.010	0.012	0.019
(7, 8)	0.060	0.159	0.079	0.154	0.059	0.154	0.000	0.404	0.000	0.345	0.001	0.052
(7, 9)	0.057	0.111	0.085	0.102	0.054	0.102	0.000	0.233	0.000	0.177	0.004	0.056
(7, 10)	0.053	0.086	0.055	0.075	0.045	0.079	0.602	0.001	0.432	0.002	0.001	0.019
(7, 11)	0.073	0.121	0.072	0.109	0.063	0.109	0.903	0.000	0.674	0.000	0.006	0.023
(7, 12)	0.009	0.108	0.013	0.079	0.007	0.082	0.244	0.002	0.189	0.010	0.001	0.018
(8, 9)	0.055	0.042	0.064	0.046	0.053	0.041	0.100	0.009	0.062	0.007	0.019	0.017
(8, 10)	0.113	0.112	0.082	0.084	0.097	0.092	0.962	0.001	0.870	0.004	0.020	0.016
(8, 11)	0.198	0.158	0.112	0.121	0.140	0.134	0.998	0.000	0.950	0.000	0.040	0.016
(8, 12)	0.059	0.095	0.033	0.067	0.047	0.072	0.823	0.006	0.705	0.028	0.014	0.025
(9, 10)	0.089	0.104	0.069	0.085	0.079	0.091	0.925	0.002	0.857	0.007	0.020	0.013
(9, 11)	0.152	0.153	0.102	0.120	0.123	0.129	0.989	0.000	0.930	0.000	0.033	0.020
(9, 12)	0.043	0.122	0.026	0.073	0.034	0.076	0.699	0.005	0.629	0.012	0.012	0.029
(10, 11)	0.076	0.051	0.071	0.054	0.074	0.051	0.240	0.006	0.189	0.004	0.048	0.037
(10, 12)	0.050	0.078	0.055	0.078	0.055	0.080	0.013	0.140	0.018	0.085	0.027	0.056
(11, 12)	0.070	0.135	0.079	0.110	0.071	0.119	0.000	0.463	0.000	0.324	0.026	0.065

4.6 Conclusions

This study evaluated CVIs in controlled simulations and in two real-world settings. The evidence supports a nuanced but consistent picture of SK performance, while identifying clear boundary conditions where classical indices can be competitive.

In multivariate normal data with location shifts, SK indices—most often SK_{std} and frequently SK_{max} —achieved the highest AAC across many (d, ρ) combinations; however, at high dimension with weak correlation, classical indices (notably DBI, and occasionally CHI) sometimes matched or surpassed SK, indicating a specific corner case where ratio-based separation can be advantageous (Table 4.1). For spread changes under the normal model, SK indices (especially SK_{max}) dominated the AAC across the reported (d, ρ) settings (Table 4.2). In the multivariate uniform model, SK_{max} was best across all displayed dimensions for both location and spread changes, reinforcing generality beyond Gaussian assumptions (Tables 4.3, 4.4). Overall, SK remains consistently strong across correlation regimes, with advantages most pronounced at moderate–high ρ , and with identifiable high- d , low- ρ cases in normal location shifts where DBI/CHI can be competitive.

For the Palmer Penguins dataset, all indices concur that pairs involving Gentoo exhibit the strongest separability: SK_{max} and SK_{std} attain the largest values, CHI and DI are also high, and DBI is relatively low in those rows, while Adélie–Chinstrap (for DBI) is the most separable pairing (Table 4.5). This cross-index agreement strengthens the conclusion that SK captures genuine structure and is not merely over-sensitive in these data.

In the mHealth streaming analysis with short batches, sedentary postures (L1–L3) are cleanly separable by SHI, DI, CHI, and SK_{max} (near-perfect values), whereas DBI shows values near zero, indicating weak DBI performance under the adopted performance scaling (Table 4.7). Static–dynamic and many dynamic–dynamic pairs remain separable yet show directional asymmetries and more conservative SHI/DI/CHI/DBI magnitudes than SK_{max} , which frequently attains values close to one in at least one ordering (Table 4.7). Kinematically similar dynamics (for example, walking versus stairs, or jogging versus running/jumping) are the most confounded by overlap-sensitive indices, although SK_{max} often identifies a strong separating direction in one of the two orderings (Table 4.7).

The complementary behavior of SK_{max} and SK_{std} adds diagnostic value beyond single-index use. Across the simulation studies and the mHealth results, SK metrics provide robust separability signals over a wide range of conditions, while acknowledging a specific boundary such as high-dimensional or weak-correlation normal location

shifts, where CHI or DBI can be competitive. Within these empirically established bounds, SK indices offer reliable, correlation-robust signals and remain compatible with short-batch operation in streaming contexts.

4.7 Summary

This chapter investigated CVIs for real-time and batch monitoring. The work integrated theoretical grounding, simulation-based evaluation, and validation on authentic datasets to assess detection capability under distributional shifts, dimensional growth, and correlation structure variation.

The chapter first outlined the challenges of streaming analytics—continuous arrival, strict memory and latency budgets, and concept drift—motivating hybrid procedures that align clustering with MSPM. A formal review positioned SHI, DI, CHI, DBI, and the proposed SK indices within a common geometric framework. The SK metrics were defined to capture maximal directional separation and the stability of separations across directions, tailored for short-batch streaming operation.

A unified MSPM–CUI framework was then described: clustering organizes incoming observations, distance-based summaries generate monitoring statistics, and CVIs provide ongoing quality signals to adapt to evolving dependence and dispersion. This design supports continuous assessment without expensive recomputation.

Extensive simulations examined location shifts and spread changes across multiple dimensions, correlations, and distributions. For multivariate normal location shifts, SK (often SK_{std} , frequently SK_{max}) achieved the highest AAC in most (d, ρ) settings, with identifiable boundary cases at high dimension and weak correlation where DBI and sometimes CHI matched or exceeded SK (Table 4.1). For spread changes in the normal model, SK (especially SK_{max}) dominated across the reported conditions (Table 4.2). Under multivariate uniform distributions, SK_{max} was best across dimensions for both shift types (Tables 4.3, 4.4). Overall, SK performance remained strong across correlation regimes, with advantages most pronounced at moderate-to-high correlation, and with specific high- d , low- ρ corners where classical indices were competitive.

Real-world evaluations corroborated these patterns. In species discrimination (Palmer Penguins), most indices indicated strongest separability for pairs involving Gentoo; SK_{max} and SK_{std} attained the largest values, with CHI/DI also high and DBI minimal, whereas Adélie–Chinstrap was least separable (Table 4.5). In mHealth streaming activity recognition with short batches, sedentary postures were nearly perfectly separated across indices; static–dynamic and several dynamic–dynamic pairs

remained separable but exhibited directional asymmetries and more conservative SHI/DI/CHI/DBI magnitudes relative to SK_{max} , particularly for kinematically similar dynamics (e.g., walking vs. stairs; jogging vs. running/jumping) (Table 4.7).

Three conclusions follow. First, the SK family provides robust separability signals in both simulations and authentic data, with SK_{max} emphasizing maximal directional contrast and SK_{std} summarizing dispersion-level stability. Second, no single index is uniformly optimal: high-dimensional, weakly correlated normal location shifts create a boundary where DBI/CHI can be competitive, while SK is consistently strongest for spread changes and under uniform models. Third, effective practice benefits from complementary indices: SK as primary signals in low-to-moderate dimensions across varying correlations, augmented by classical indices in high- d , low- ρ cases to hedge edge conditions.

These contributions include (i) an SK-based metric family designed for streaming constraints, (ii) an MSPM–CVI integration enabling continuous, correlation-robust assessment, and (iii) an evidence-backed delineation of regimes where SK offers clear advantages versus where classical indices remain competitive. Together they support responsive, data-driven monitoring under dynamic operational conditions while clarifying context-specific boundaries on index effectiveness.

Chapter 5

Monitoring Applications in Maritime Data: Environmental Profiling and Engine Classification

5.1 Introduction

The assessment of ships' environmental performance through M, as presented in the current chapter, can be closely related to RTM workflows by integrating continuous data streams from onboard sensors and operational systems. Real-time environmental monitoring systems capture dynamic data such as fuel consumption, Carbon Dioxide CO₂ emissions, engine status, and navigational parameters during ship voyages.

By applying M techniques like PCA and clustering to these ongoing data streams, it becomes possible to generate real-time environmental performance indices that reflect the current operational efficiency and compliance status of vessels. This approach not only supports timely decision-making and corrective actions to optimize energy use and reduce emissions but also enables adaptive monitoring that can detect and respond to environmental performance drift over time, thus enhancing sustainability and regulatory compliance in maritime operations.

Furthermore, such RTM efforts are enabled by robust sensor networks and data acquisition platforms that continuously record environmental, mechanical, and navigational parameters, often under harsh sea conditions. M models can leverage this rich, continuous information to handle challenges such as missing or noisy data through imputation and dimensionality reduction, ensuring reliable and interpretable insights. By detecting subtle changes or shifts in performance clusters, these systems can forecast

maintenance needs, optimize operational parameters, and support regulatory reporting with minimal delay.

In practice, integrating M-based environmental performance indices with real-time data streams creates a feedback loop whereby vessels dynamically adapt to evolving operational and environmental conditions. This integration positions shipping companies to reduce greenhouse gas emissions, improve fuel efficiency, and meet tightening global regulations, all while maintaining operational efficiency. The continuous, adaptive nature of monitoring (see Figure 5.1) supports strategic decision-making for fleet management, enabling data-driven responses that promote safer, greener, and more cost-effective maritime transportation.

This formulation is informed by recent maritime studies that integrate operational telemetry with M for environmental performance assessment and compliance monitoring [16, 17].

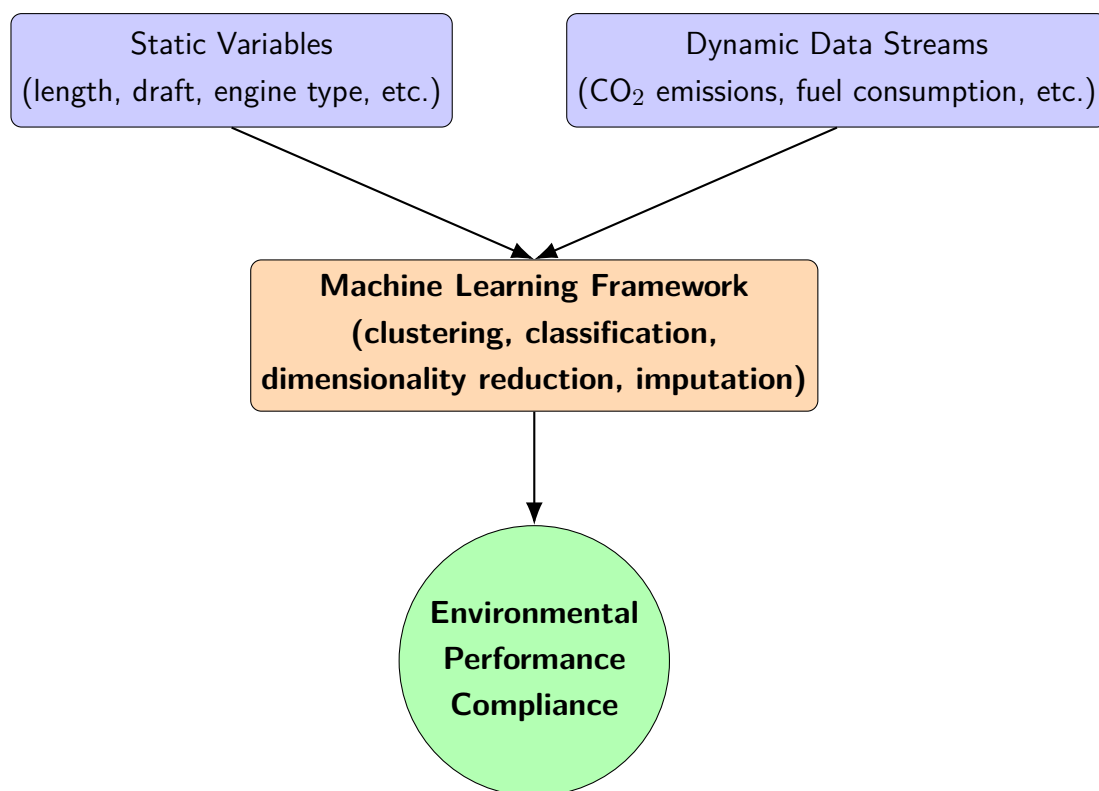


Fig. 5.1 A conceptual diagram of the monitoring procedure for assessing a ship's environmental compliance. Static and dynamic variables are fed into a ML framework to produce a final performance and compliance status.

5.2 Motivation

The maritime shipping industry faces unprecedented environmental challenges that demand immediate attention and innovative approaches to performance monitoring. Ships account for over 80% of world trade [156], yet greenhouse gas emissions from maritime transport rose 9.6% from 2012 to 2018 [157] and are projected to increase significantly by 2050 [158]. This alarming trajectory threatens global climate targets and necessitates fundamental changes in how the industry monitors, assesses, and optimizes vessel environmental performance.

The International Maritime Organization (IMO) and European Commission (EC) have responded with increasingly stringent regulatory frameworks. The IMO introduced the OILPOL Convention in 1954 and continues to enforce comprehensive measures including the Energy Efficiency Design Index (EEDI), Energy Efficiency Operational Indicator (EEOI), and Ship Energy Efficiency Management Plan (SEEMP), focusing on reducing emissions through design and operational strategies [158, 159, 160]. The EC promotes initiatives supporting emission reductions and maritime decarbonization targets, including integrating maritime transport into emissions trading [161, 162]. These regulations demand accurate, data-driven assessment of vessel environmental performance and technical specifications through continuous monitoring systems that can provide real-time compliance verification. However, the tools and methodologies currently available to the industry remain inadequate to meet these escalating demands.

The digitization of shipping operations has fundamentally transformed the maritime landscape. Modern vessels are equipped with sophisticated sensor networks that continuously capture dynamic data streams including fuel consumption rates, CO₂ emissions concentrations, engine operational parameters, vessel speed, navigational conditions, and ambient environmental factors during voyages. While data monitoring equipment costs are relatively low [163], effectively processing these continuous data streams remains extraordinarily complex due to multiple challenges. Variability in ship activities creates inconsistent data patterns [164]. Sensor malfunctions, communication interruptions, and maintenance periods introduce gaps in monitoring continuity. Incomplete measurements, noise, and temporally dependent patterns require sophisticated preprocessing before meaningful analysis becomes possible. The maritime industry possesses vast quantities of operational data but lacks robust frameworks to transform this raw information into actionable environmental performance insights that support real-time decision-making.

Traditional approaches to evaluating vessel environmental performance have relied primarily on resistance modeling, which estimates fuel consumption and emissions

based on theoretical calculations of vessel resistance influenced by speed and external factors such as wind and waves [165, 166]. While foundational, these static theoretical models struggle with the complexities inherent in dynamic real-world operations, including variable weather conditions, diverse loading states, operational practices varying by crew and company, and equipment degradation over time. The disconnect between theoretical predictions and observed operational performance creates uncertainty in environmental assessments and limits the effectiveness of optimization strategies. Research has established that vessel speed is the principal determinant of fuel consumption, with weather-related resistance and loading conditions exerting notable secondary influences [167, 168], yet translating this knowledge into practical monitoring systems that account for all operational variables remains challenging.

Despite substantial research interest in maritime environmental sustainability, critical gaps persist in current monitoring and assessment approaches. First, maritime operational datasets invariably contain significant deficiencies including missing values, duplicates, and measurement errors resulting from sensor malfunctions, human errors, data transmission failures, and maintenance periods. These data quality issues undermine the reliability of environmental assessments and hinder effective real-time decision-making, leading to operational uncertainty and increased regulatory risk. The challenge of maintaining continuous, high-quality data streams from vessels operating in diverse and often harsh maritime environments represents a fundamental obstacle to comprehensive environmental monitoring.

Second, essential technical specifications, particularly engine model information, are frequently missing or unreliable in maritime databases despite being fundamental to accurate performance assessment. The engine model determines fuel consumption characteristics, emissions profiles, operational efficiency baselines, and expected performance under various loading and speed conditions. However, this information is often absent due to privacy concerns, commercial sensitivities surrounding proprietary vessel designs, limited data sharing across commercial platforms, and inconsistent reporting practices. Without accurate engine specifications, environmental assessments rely on generic assumptions that may poorly reflect actual vessel performance, compromising both compliance verification and optimization efforts.

Third, existing environmental performance indices and assessment frameworks exhibit significant limitations. Initiatives such as the Clean Shipping Index (CSI), Environmental Ship Index (ESI), and Rightship's Existing Vessel Design Index (EVDI) have emerged to encourage environmental performance improvements. However, these approaches often focus narrowly on specific technical aspects such as alternative fuel

adoption [169], fuel life cycle analyses [170], hull maintenance [167], or vessel design features [171, 172], without providing comprehensive real-time operational assessment that integrates multiple performance dimensions. Comparative studies reveal limitations in these indices' ability to accurately evaluate dynamic performance [173], and concerns persist about their contribution to mitigating maritime greenhouse gas emissions [174]. The shipping industry faces numerous, sometimes contradictory environmental initiatives that complicate decision-making and increase administrative burdens [164], yet no standardized, comprehensive framework exists for continuous environmental performance monitoring that can guide operational decisions in real time.

Fourth, the temporal dimension of environmental monitoring presents unique challenges. Vessels operate continuously across varying conditions, encountering different weather patterns, sea states, loading configurations, and operational demands. Environmental performance fluctuates dynamically in response to these changing conditions, yet most assessment frameworks provide only static, periodic evaluations that fail to capture this temporal variability. The inability to monitor environmental performance continuously and respond adaptively to changing conditions limits the maritime industry's capacity to optimize operations for emissions reduction while maintaining schedule reliability and operational efficiency.

Finally, the integration of environmental performance assessment with operational decision-making remains underdeveloped. Even when monitoring systems capture relevant data and generate performance metrics, these insights often fail to translate into timely operational adjustments that could reduce emissions and improve efficiency. The gap between data collection, analysis, and actionable recommendations creates missed opportunities for real-time optimization. Vessel operators require not only accurate assessments of current environmental performance but also clear, interpretable guidance on how specific operational changes such as speed adjustments, route modifications, or maintenance interventions could improve outcomes. The absence of integrated monitoring and decision support frameworks represents a critical barrier to achieving the maritime industry's environmental objectives.

The urgency of addressing these challenges intensifies as regulatory requirements tighten, public awareness of maritime environmental impacts grows, and the industry faces increasing pressure from stakeholders including port authorities, cargo owners, investors, and environmental organizations. Shipping companies must navigate this complex landscape while maintaining economic viability in a highly competitive global market. The need for robust, reliable, data-driven environmental monitoring systems that can operate continuously in real-world conditions, handle incomplete and noisy

data, infer missing technical specifications, provide comprehensive performance assessment across multiple dimensions, and support immediate operational decision-making has never been more critical. Developing such systems requires advancing beyond traditional theoretical models and current fragmented assessment approaches to leverage modern ML techniques and real-time data analytics in ways specifically tailored to maritime operational realities.

5.3 Assessing Environmental Performance of Vessels

This chapter introduces environmental performance indices derived by combining clustering algorithms with PCA. Analyzing 14 operational and design variables, ships are grouped into four clusters reflecting their physical dimensions, operating region, and operational environmental efficiency, revealing distinct characteristics within each group. Environmental performance of ships is a vital concern in the shipping industry, driven by climate change and evolving global regulations. The digitization of shipping has resulted in the collection of large amounts of environmental performance data during voyages, presenting opportunities to develop data-driven performance models using ML.

Maritime data often contain deficiencies such as missing values, duplicates, and errors due to human or system limitations. These issues hinder effective decision-making, leading to reduced productivity and increased costs. Missing critical information like the vessel's primary engine model significantly impacts its operational and environmental performance. To address this, an analysis and comparison of several ML classification methods were conducted to identify the main engine model of ships. Imputation methods manage missing data, while dimensionality reduction techniques such as PCA, UMAP, and t-SNE are combined with classifiers to optimize performance. The best results were achieved using ExtraTreeClassifier combined with these dimensionality reduction methods.

While engine model classification is a one-time process per ship, the integration of data streaming enables continuous updating of environmental performance indices. Imputation plays a crucial role in handling missing or flawed operational data, ensuring reliable, near-RTM. This dynamic system provides immediate insights and recommendations, allowing ship operators to adjust operations sustainably throughout a voyage. Ships have long dominated goods transportation, accounting for over 80% of world trade [156]. The drive to reduce environmental pollution has led to new regulatory

measures aimed at enhancing energy and operational efficiency. Despite these efforts, greenhouse gas emissions rose 9.6% from 2012 to 2018 [157] and are projected to increase significantly by 2050 [158].

The IMO introduced the OILPOL Convention in 1954 and continues to enforce measures such as the EEDI, EEOI, and SEEMP, focusing on reducing emissions through design and operational strategies [159, 160, 158]. Speed reduction, a widely used and cost-effective measure, substantially lowers greenhouse gas emissions [175, 176]. The EC also promotes initiatives supporting emission reductions and maritime decarbonization targets, including integrating maritime transport into emissions trading [161, 162]. Several indices, such as CSI, ESI, and Rightship's EVDI, encourage environmental performance improvements.

While data monitoring equipment costs are relatively low [163], data processing remains complex due to variability in ship activities [164]. Performance modeling includes theory-based designs with operational uncertainties [177], alongside emerging data-driven approaches fueled by digitization and ML [178].

Application of ML includes neural networks estimating shaft power [179, 180, 181, 182], Bayesian networks quantifying port energy performance [183], and regression comparisons predicting fuel consumption [184, 185]. Commercial tools leveraging these methods have launched, although interpretability and sensitivity remain challenging [186]. As regulatory pressures increase, standardized, data-driven environmental performance metrics become imperative. This chapter contributes useful indices integrating design and operational data to aid comprehensive ship performance assessment. The following sections detail the theoretical background, methods, analysis, results, and conclusions of this work.

5.3.1 Theoretical Background

Sustainable transformation in the maritime sector has long been a priority, yet it entails complex decisions involving multiple, often conflicting factors [187]. Efforts to improve the environmental performance of vessels while maintaining service levels frequently face trade-offs [188], leading many management systems to prioritize cost or operational performance metrics [189].

Despite substantial research interest in environmental sustainability, many proposed solutions remain theoretical or impractical. The shipping industry is challenged by numerous, sometimes contradictory environmental initiatives that complicate decision-making and increase administrative burdens [164]. Many current approaches focus

narrowly on technical aspects, such as alternative fuels [169], fuel life cycle analyses [170], hull cleaning [167], or vessel design [171, 172].

Several initiatives offer performance indicators based on environmental factors, while others are certification or incentive schemes that reward vessels or practices with environmental improvements, potentially offering market advantages [161]. Some target specific issues, uses, locations, or vessel types, whereas others assess a wider range of environmental factors. Nonetheless, the overall effectiveness of these measures in enhancing environmental performance is debated. For example, comparative studies of the CSI and ESI reveal limitations in evaluating performance accurately [173]. Furthermore, concerns about the opacity and ambition of private standards raise questions about their contribution to mitigating maritime GreenHouse Gas (GHG) emissions [174].

Modeling ship fuel consumption and emissions has traditionally relied on “resistance modeling”, which estimates total vessel resistance based on speed and external factors like wind and waves [165, 166]. However, this approach struggles with complexities inherent in real-world operations, prompting development of alternative methods [168, 190, 191, 192, 193]. Research consensus holds that vessel speed is the principal determinant of fuel consumption, although weather-related resistance exerts a notable influence [167, 168].

Achieving accurate and interpretable assessments of environmental performance depends heavily on approach, data complexity, and utilization of raw data. Artificial Neural Networks (ANNs) have successfully predicted propulsive power based on key factors such as speed, wind speed, direction, and temperature [194]. Other empirical studies have employed data sources like noon reports [167, 168, 195] and Automatic Identification System (AIS) vessel position data [196]. The effectiveness of ANN-based fuel consumption models is affirmed by studies focusing on individual vessels, such as oil tankers [197], with ANN models often outperforming traditional polynomial regression and SVMs techniques [198].

This chapter proposes an alternative, ML-based approach for assessing ship environmental performance, emphasizing objective and quantified evaluation.

5.3.2 Materials and Methods

This work employs a ML framework to develop composite energy efficiency indices based on real operational data from ships (see Figure 5.2 for a simplified overview). The framework integrates PCA and clustering techniques to extract meaningful combined

indices from real data, aiming to reduce the number of parameters needed to characterize a ship's environmental performance.

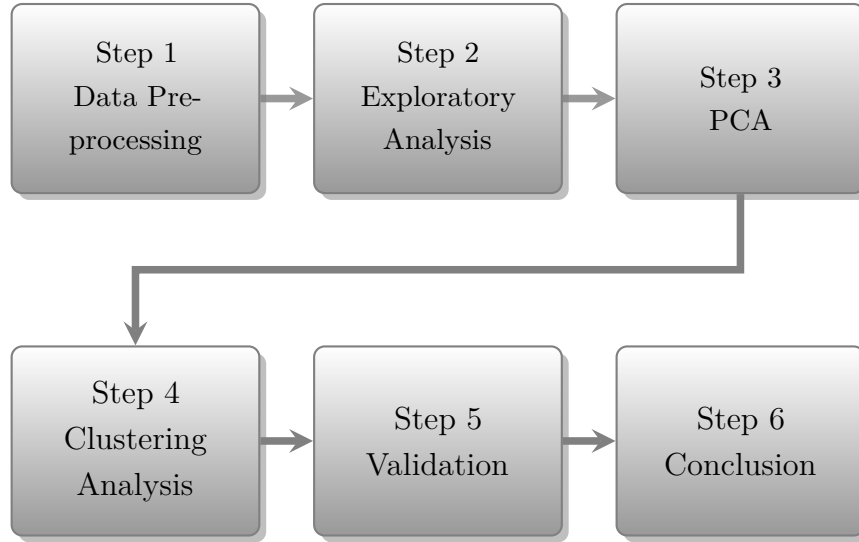


Fig. 5.2 Framework of the analysis process.

Optimally, a single artificial environmental performance index would summarize most of the information contained within the data. However, due to the complexity and diversity of the variables, it is often more informative to extract multiple indices that capture different aspects (e.g., pollution levels and pollution sources), enabling practical and nuanced interpretations.

To derive such indices, PCA is first applied to reduce data dimensionality while preserving relationships among variables. Subsequently, CA categorizes ships with similar environmental performance profiles. PCA is widely recognized in industrial applications [42] for effective dimension reduction and has a proven history in developing environmental performance indices across various sectors [199, 200].

Principal Component Analysis

PCA is a widely used mathematical technique [201] that transforms a set of potentially correlated variables into a smaller number of uncorrelated variables called Principal Components (PCs), without making assumptions about the underlying distribution of the data. This orthogonal transformation reveals the internal data structure by explaining maximum variance in the data through the first few components. Formally, given p original variables X_i , $i = 1, 2, \dots, p$, each principal component Y_j can be

expressed as a linear combination:

$$Y_j = a_{j1}X_1 + a_{j2}X_2 + \cdots + a_{jp}X_p,$$

where the coefficients a_{ju} represent the contribution weights of each original variable to the component. PCA is performed by decomposing the covariance matrix S (or correlation matrix P for standardized data) of the variables, extracting components that are orthogonal (uncorrelated) to each other. Typically, the first few PCs capture the majority of the variance, allowing dimensionality reduction. In many datasets, including those with a limited number of variables such as this study, the first two or three PCs often summarize most of the information.

By the Central Limit Theorem [202], each PC, as a sum of random variables, approximates a Normal distribution, enabling assumptions of independence when the PCs are uncorrelated. This simplification facilitates interpreting the multivariate data as a set of independent univariate components. In the context of this work, PCA serves as an essential tool to reduce complexity and extract meaningful indices representing environmental performance dimensions, aiding subsequent clustering and monitoring analyses.

Cluster Analysis

CA encompasses a variety of algorithms designed to group a set of observations so that members within each group share similar characteristics, while differing from those in other groups. As a fundamental tool for exploratory data analysis, clustering is widely applied across multiple scientific disciplines to reveal inherent data structures. Different clustering algorithms interpret clusters in diverse ways—some focus on grouping data points based on spatial proximity, density, or statistical distribution. This variety frames clustering as a multi-objective optimization problem where the selection of the appropriate method and parameters (e.g., distance metric, number of clusters, density thresholds) depends on the dataset's nature and the objectives of the analysis.

Similarity measures, often expressed through distance functions such as Euclidean, Manhattan, Minkowski, or Cosine similarity, form the basis of most clustering approaches. The goal is to construct homogeneous groups (clusters) with strong internal similarity. Importantly, clustering is typically not a one-shot process but involves iterative refinement. Analysis may require multiple runs with adjusted preprocessing steps and model parameters until clusters manifest desirable properties, enabling reliable interpretations aligned with environmental monitoring goals.

Available Real Data

To develop novel environmental performance indices for ships, datasets from two principal sources were merged. The first is the EU Monitoring, Reporting, Verification (MRV) mechanism, which systematically collects CO₂ emissions data for ships above 5000 gross tonnage operating in ports under the jurisdiction of any EU Member State. The second dataset originates from the startup “27 Research” in Greece, providing detailed physical characteristics for ships with at least one voyage in the EU zone between 2018 and 2021. The final consolidated dataset comprises two main categories: General Cargo ships (2650 records) and Container ships (62 records). Table 5.1 summarizes the recorded variables for each vessel.

Table 5.1 Data Set Description.

Variable	Description
X_1	Annual total fuel consumption for voyages (million tonnes)
X_2	Aggregated CO ₂ emissions from voyages between ports under Member State jurisdiction (million tonnes)
X_3	Aggregated CO ₂ emissions from voyages departing ports under Member State jurisdiction (million tonnes)
X_4	Aggregated CO ₂ emissions to ports under Member State jurisdiction (million tonnes)
X_5	CO ₂ emissions within ports under Member State jurisdiction at berth (million tonnes)
X_6	Annual total time spent at sea (hours)
X_7	Total distance traveled (nautical miles)
X_8	Deadweight tonnage (tonnes)
X_9	Draft; distance between waterline and keel (meters)
X_{10}	Maximum length perpendicular to waterline (meters)
X_{11}	Maximum width (beam) (meters)
X_{12}	Depth from keel bottom to main deck (meters)
X_{13}	Maximum available cargo space (cubic meters)
X_{14}	Engine power (kilowatts)

5.3.3 Analysis and Results

The subsequent analysis utilizes primarily PCA and CA as detailed in Section 5.3.2, following essential preprocessing and exploratory data analysis steps to identify and exclude outliers.

Data Preprocessing

Prior to PCA, each variable X_i undergoes standardization using:

$$Z_i = \frac{X_i - \mu_i}{\sigma_i},$$

where μ_i and σ_i are the mean and standard deviation of the i -th variable. When population parameters are unknown, unbiased sample estimates are employed to approximate these values.

This preprocessing ensures variables are comparable in scale, enhancing the efficacy and interpretability of the PCA and clustering steps.

Exploratory Analysis of the Original Data

Following data preprocessing, 381 records with missing values were excluded, leaving 2209 complete observations for analysis. An initial exploratory data analysis was performed to examine correlations among variables and detect potential outliers. This step is crucial for justifying the application of PCA by confirming the presence of highly correlated variables and improving PCA results through outlier removal, thereby reducing noise in the multivariate dataset. Significant positive correlations emerged, notably between fuel consumption (X_1) and variables associated with environmentally detrimental emissions ($X_2, X_3, \dots, X_5$). Strong correlations were also observed among ship construction characteristics ($X_8, X_9, \dots, X_{14}$). Figure 5.3 displays a correlation heatmap highlighting significant Pearson correlation coefficients within the dataset. Further inspection revealed that total fuel consumption is correlated with total distance traveled (X_7), which itself correlates with annual time at sea (X_6). Among technical ship specifications, very strong positive correlations ($r \geq 0.9$) exist between deadweight (X_8) and both beam (X_{11}) and cargo volume (X_{13}). Cargo volume also exhibits strong correlation with vessel length (X_{10}) and beam (X_{11}).

Outlier detection concentrated on ship characteristics (X_8 to X_{14}), employing univariate graphical analyses alongside multivariate Mahalanobis Distance (MD) measures. MDs quantify the distance of each observation from the multivariate mean, given by $\sqrt{(u - \mu)V^{-1}(u - \mu)^T}$, where u is the observation vector, μ the mean vector, and V the covariance matrix. Figure 5.4 illustrates MD distribution for observations.

Most observations fall below an MD threshold of 10, with a minority exceeding this value. An upper MD limit of 10 was employed to exclude probable erroneous entries. For instance, a vessel with reported draft (X_9) of 81.15 m greatly exceeded the mean value ($\bar{x}_9 = 8.30$ m, standard deviation $sd_{X_9} = 3.32$ m), indicating a data

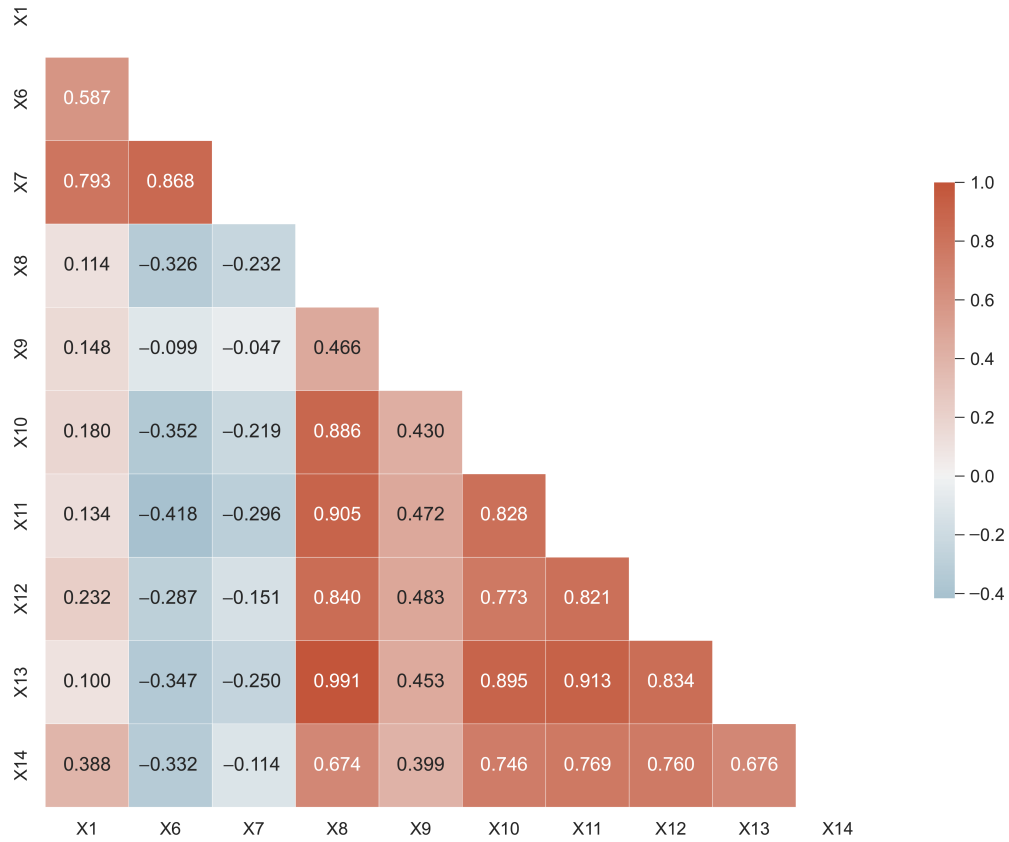


Fig. 5.3 Correlation matrix of dataset.

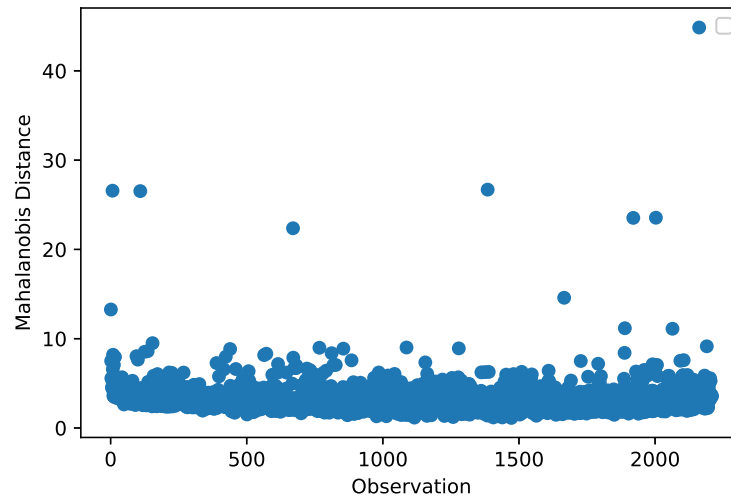


Fig. 5.4 Mahalanobis Distance and Outlier Detection.

error. Similarly, an outlier with cargo space of 62400 m^3 (over 3 standard deviations above the mean $\bar{x}_{13} = 19452 \text{ m}^3$, $sd_{X_{13}} = 13265.28 \text{ m}^3$) was removed.

Application of PCA

PCA was conducted sequentially on two datasets: (a) the complete dataset and (b) a reduced dataset from which observations (ships) with a Mahalanobis Distance (MD) exceeding 10 were excluded. Determining the number of principal components to retain was guided primarily by the scree plot analysis.

The scree plot in Figure 5.5 depicts eigenvalues against component indices, highlighting a sharp decline up to the third principal component, beyond which the curve flattens. This “elbow” indicates that retaining three principal components captures the majority of data variance. Notably, this cutoff was consistent in both datasets.

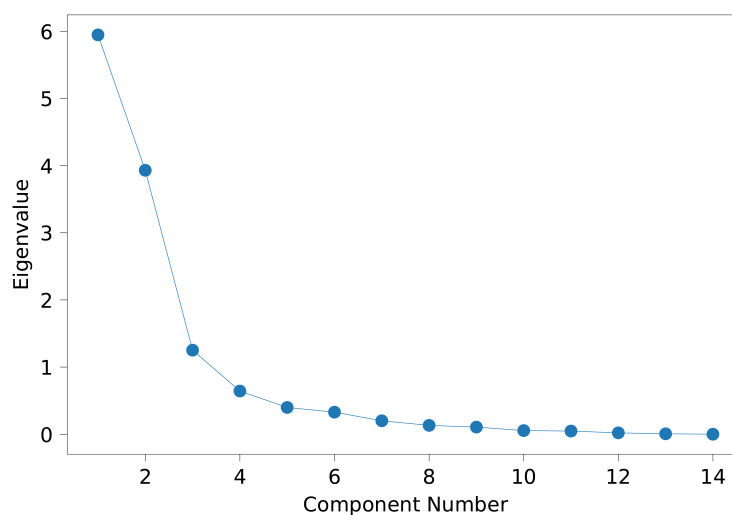


Fig. 5.5 Scree plot for principal components.

Table 5.2 presents the loadings of the first three principal components for both the initial standardized dataset and the reduced dataset. The similarity of loadings across datasets is reflected by minimal differences in the "Pairwise Differences" columns. The first three PCs explain over 80% of the variance in both cases (80.2% and 85.1%, respectively).

Most variables, except Z_2 , Z_7 , and Z_8 , show a positive correlation with PC1, associated primarily with ship physical dimensions. Variables Z_8 , Z_{10} , Z_{11} , and Z_{13} correlate negatively with PC2, which represents operational environmental efficiency. PC3 correlates negatively with variables Z_1 , Z_3 , Z_4 , Z_7 , and Z_{14} , linked to CO₂ emissions from different geographical regions.

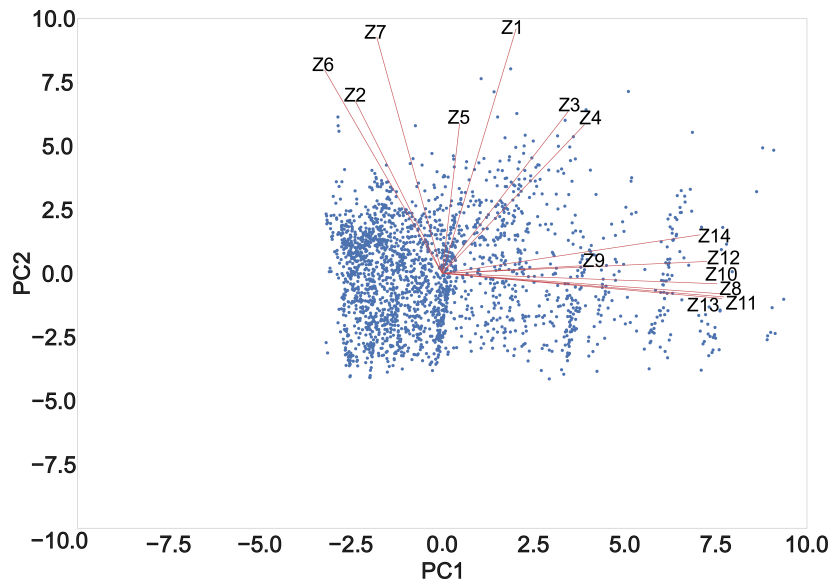


Fig. 5.6 Biplot showing variables' relation to the first two principal components (based on reduced dataset).

The biplot (Figure 5.6) visualizes these relationships, supporting clear interpretation of the three key principal components representing physical size, environmental efficiency, and area of operation.

Table 5.2 PCA loadings for the first three PCs on initial and reduced datasets with pairwise differences.

Variable	Initial Standardized			Reduced Dataset			Pairwise Differences		
	PC1	PC2	PC3	PC1	PC2	PC3	PC1	PC2	PC3
Z ₁	0.099	0.478	-0.084	0.101	0.480	-0.082	0.002	0.001	0.002
Z ₂	-0.119	0.334	0.528	-0.118	0.337	0.537	0.000	0.003	0.009
Z ₃	0.170	0.317	-0.455	0.172	0.317	-0.456	0.002	0.000	0.001
Z ₄	0.192	0.291	-0.454	0.194	0.291	-0.455	0.002	0.000	0.002
Z ₅	0.022	0.296	0.430	0.024	0.293	0.408	0.002	0.003	0.022
Z ₆	-0.159	0.397	0.094	-0.161	0.398	0.102	0.002	0.001	0.007
Z ₇	-0.091	0.465	-0.003	-0.089	0.463	0.001	0.001	0.002	0.004
Z ₈	0.378	-0.040	0.196	0.382	-0.040	0.203	0.005	0.000	0.007
Z ₉	0.226	0.031	0.058	0.191	0.015	0.061	0.035	0.016	0.003
Z ₁₀	0.372	-0.020	0.068	0.376	-0.020	0.073	0.003	0.000	0.005
Z ₁₁	0.382	-0.046	0.105	0.385	-0.046	0.108	0.003	0.000	0.003
Z ₁₂	0.366	0.019	0.109	0.362	0.024	0.102	0.004	0.005	0.007
Z ₁₃	0.379	-0.049	0.186	0.382	-0.049	0.192	0.004	0.000	0.006
Z ₁₄	0.347	0.074	-0.064	0.352	0.076	-0.064	0.005	0.001	0.001
Variance Explained	0.431	0.281	0.090	0.455	0.301	0.096	0.025	0.020	0.006

Interpretation of the First Principal Components

High absolute values in PC loadings reveal variables with strong influence on that component. PC1 chiefly represents ship physical dimensions, characterized by positive loadings for variables Z_8 through Z_{14} . PC2 reflects operational environmental efficiency, indicated by negative correlations with variables Z_1 through Z_7 , related to consumption and CO₂ emissions. PC3 contrasts CO₂ emissions in different regions with positive loadings for variables Z_2 to Z_5 , marking ships operating predominantly within or outside Member State jurisdictions.

Thus, the three PCs can be labeled:

- Ship's Physical Dimensions (PC1);
- Operational Environmental Efficiency (PC2);
- Operating Region (PC3).

These independent indices collectively characterize ships by size, environmental performance, and operating region.

Regression-Adjusted Monitoring for Correlated Variables

The strong correlations observed among vessel operational variables, particularly between ballast and laden conditions for speed and fuel consumption, suggest potential benefits from regression-adjusted monitoring approaches. While PCA effectively reduces dimensionality and captures variance structure, regression-adjusted control charts [59] offer complementary advantages for real-time anomaly detection when variables exhibit correlation coefficients exceeding 0.7. By monitoring residuals from regression models rather than raw variables, such approaches can more sensitively detect subtle deviations in environmental performance that may indicate equipment degradation, operational inefficiency, or compliance drift. Integration of regression-adjusted methods with the PCA-based environmental performance indices could enhance RTM systems' ability to identify vessels requiring immediate operational adjustments or maintenance interventions, particularly when correlation structures change due to varying operational conditions or equipment aging.

Graphical Tag Definition Based on the Three Indices

To visualize ship environmental performance while accounting for vessel size and operating region, a composite graphical tag was designed (Figure 5.7). This base tag is

intended to be color-mapped in the subsequent figures: the lower zone adopts the color scale used for PC1, the upper zone adopts the green-to-red gradient used for PC2, and the stern adopts the color scale used for PC3.

The lower zone's fill proportionally reflects ship size with blue intensity; large vessels have nearly full blue lower zones, while small vessels have almost none. The upper zone color gradates from green (environmentally friendly) to red (high pollution), indicating operational environmental efficiency. The stern color communicates operational region category—red for predominantly outside Member State jurisdiction, green for inside, and orange for mixed operations.

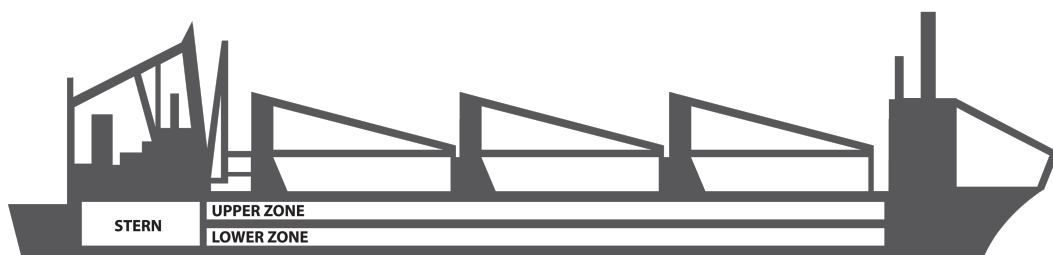


Fig. 5.7 Base graphical tag representing environmental performance.

The tag coloring follows:

- Standardize tag zone lengths to 1;
- Fill lower zone blue proportional to the empirical cumulative distribution value of PC1;
- Fill upper zone with green, orange, or red depending on scaled intervals reflecting environmental efficiency;
- Color stern based on PC3 score quartiles: red if below first quartile, green if above third, else orange.

Typical tags (Figure 5.8) illustrate variance in size, environmental efficiency, and operating region, respectively reflecting vessel clusters. These tags provide intuitive, at-a-glance environmental performance indexing, distinguishing ships with high emissions from similarly sized but cleaner vessels, as well as categorizing operational jurisdictions.

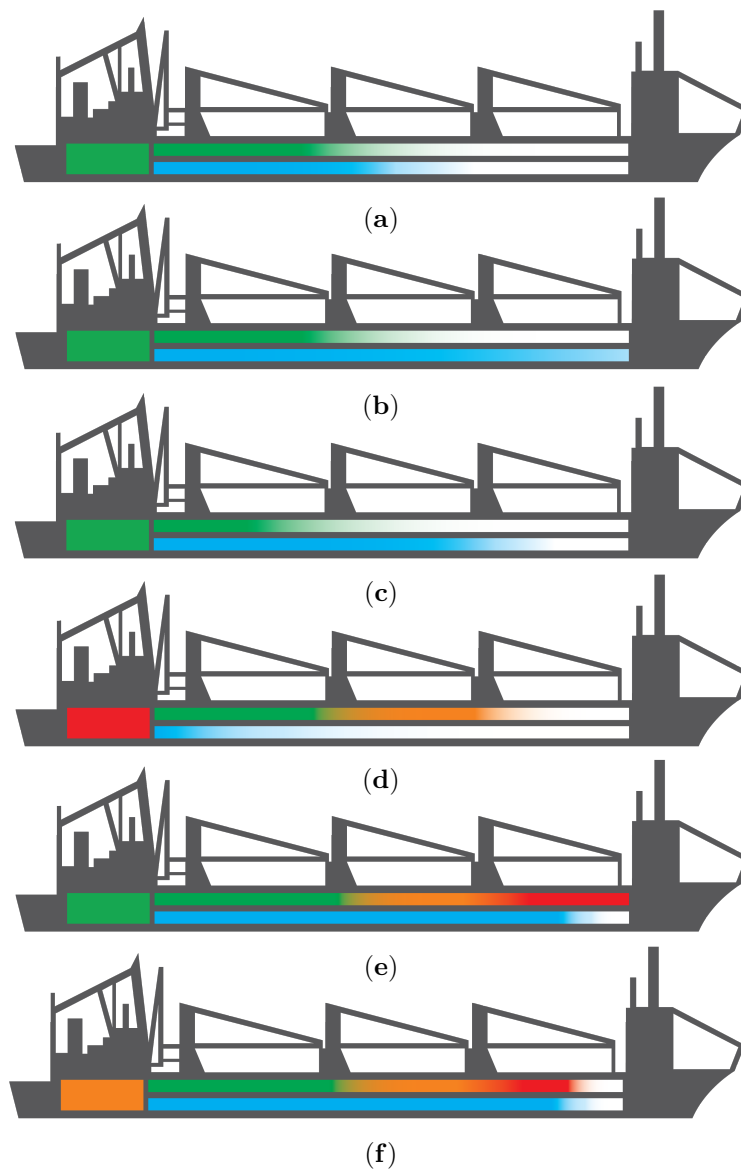


Fig. 5.8 Representative graphical tags for ship environmental performance.

This graphical tagging succinctly encapsulates CO₂ emissions adjusted by cargo capacity, thereby effectively differentiating ships by size and operational environmental impact.

Cluster Analysis Based on the Three Indices

Building upon the three principal components representing ship environmental impact—derived by merging operational time, emissions, and technical factors such as physical dimensions—the K-Means clustering algorithm was employed to identify

groups of vessels with similar characteristics. This approach clusters ships based on size, CO₂ emissions, operational time, and energy consumption as captured by the three PCA-derived indices.

Choosing the Optimal Number of Clusters

Selecting the optimal number k of clusters is critical in CA. Various criteria exist, each highlighting different data attributes and cluster qualities, which may suggest differing k values. Thus, it is advisable to evaluate multiple scenarios and integrate domain expert judgment to finalize a statistically sound and practically meaningful cluster count.

This study applies two common methods—silhouette score [9] and Gap statistic [203, 204]—to determine k using the PCA components of General Cargo ships. These metrics are calculated over a range of k values; peaks or high values indicate clusters with well-matched internal cohesion and good separation from neighboring clusters.

Figure 5.9 presents silhouette scores and Gap statistics (Figure 5.10), both peaking at $k = 4$, suggesting four clusters optimally represent the data. The silhouette score also indicates two clusters as a possible alternative, warranting examination of both configurations.

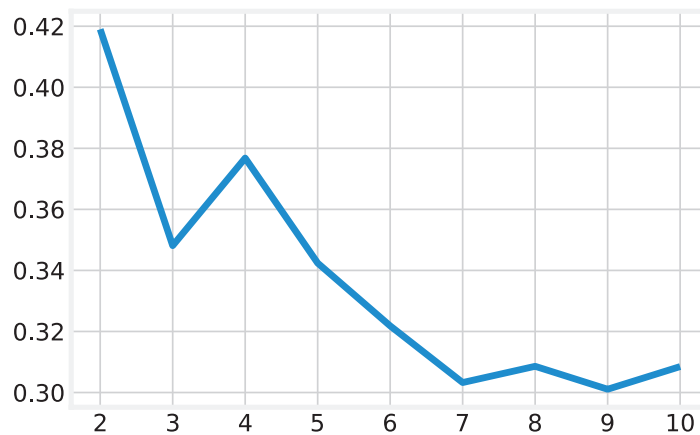


Fig. 5.9 Silhouette Score for cluster numbers analyzed by k-means on the three principal components.

K-Means Clustering Results

The k-means algorithm was executed with cluster numbers $k = 2$ and $k = 4$. Figures 5.11 through 5.13 illustrate the two-cluster solution projected onto planes defined by pairs of PCs, demonstrating that $k = 2$ clusters primarily differentiate ships by physical

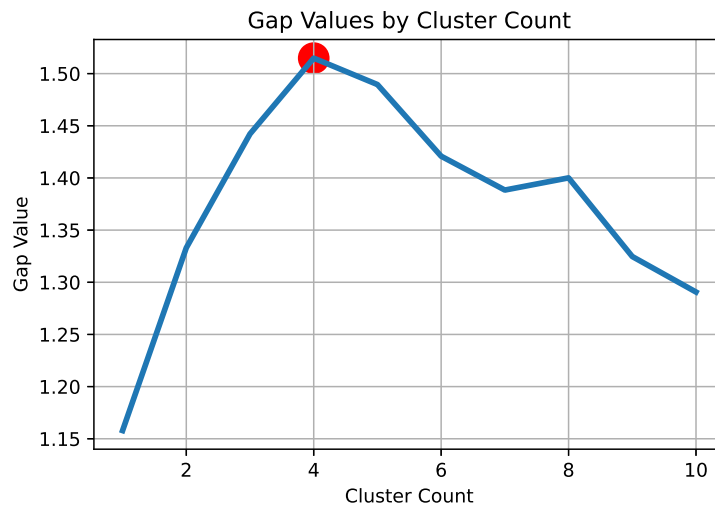


Fig. 5.10 Gap Statistic for cluster numbers analyzed by k-means on the three principal components.

size (PC1) but do not sufficiently distinguish operational efficiency or operating region (PC2 and PC3).

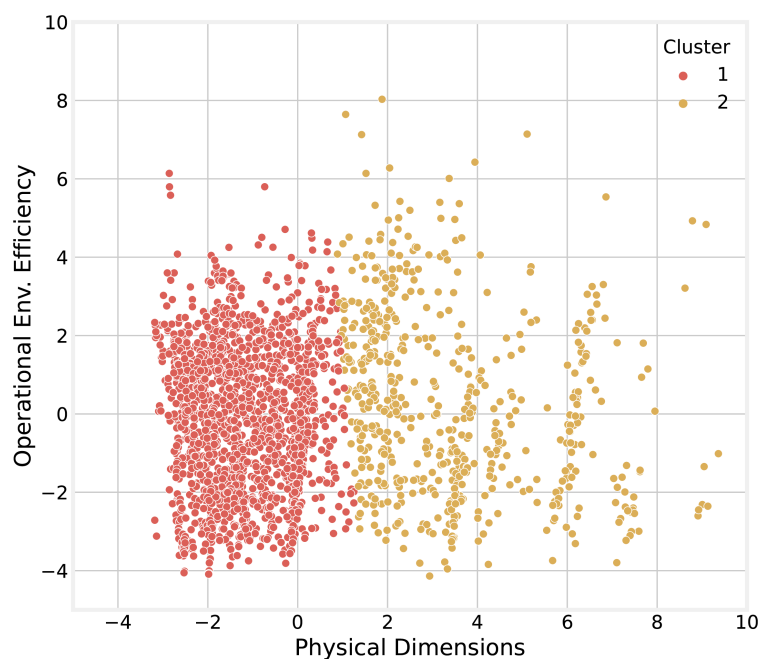


Fig. 5.11 Two clusters on the plane defined by first and second PCs.

Conversely, Figures 5.14 through 5.16 exhibit the four-cluster configuration, revealing distinct groupings that differentiate not only ship size (PC1) but also operational

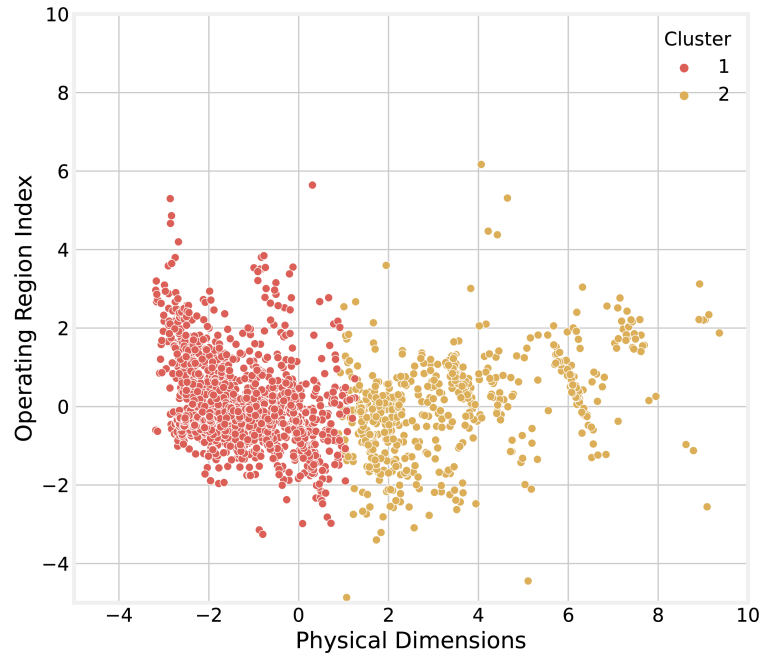


Fig. 5.12 Two clusters on the plane defined by first and third PCs.

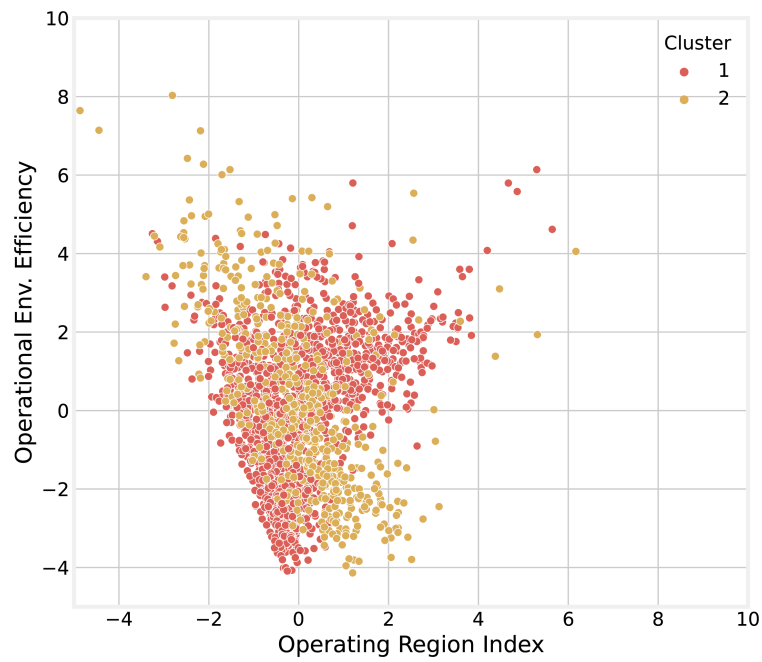


Fig. 5.13 Two clusters on the plane defined by second and third PCs.

environmental efficiency (PC2). The operating region (PC3) contributes less prominently to the cluster separation.

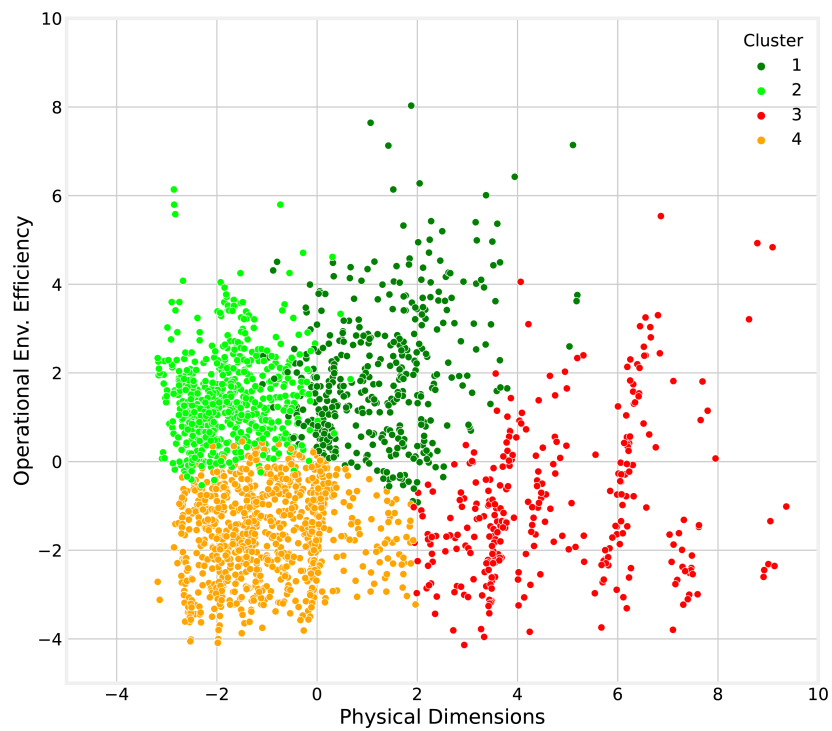


Fig. 5.14 Four clusters on the plane defined by first and second PCs.

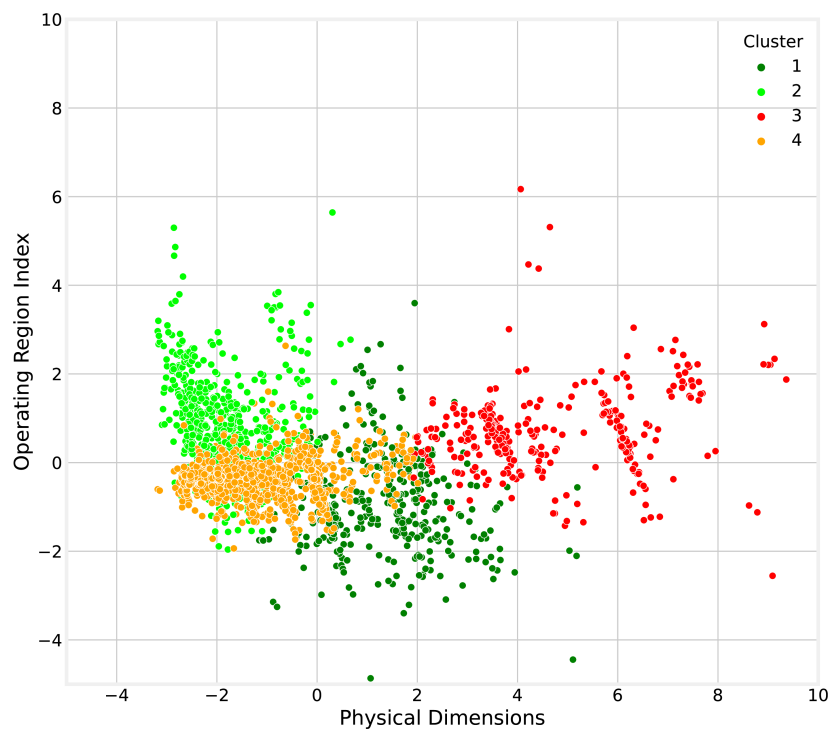


Fig. 5.15 Four clusters on the plane defined by first and third PCs.

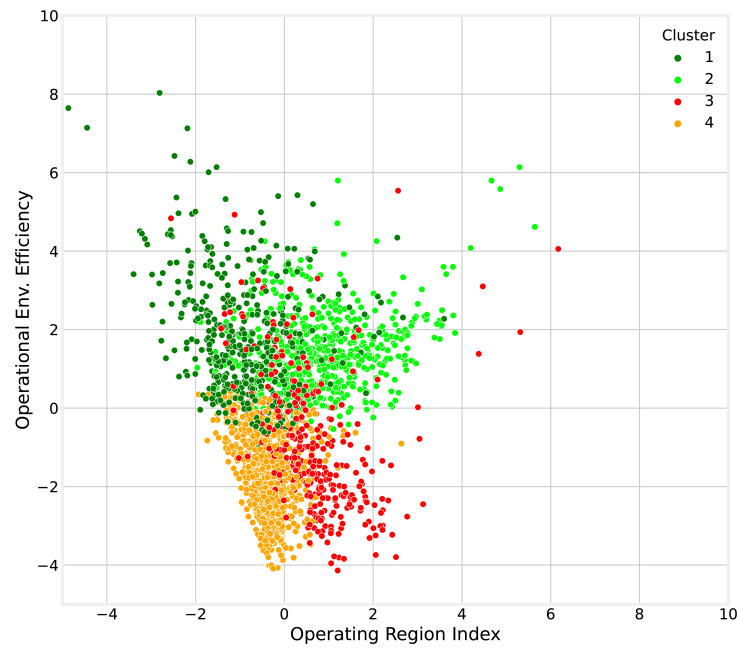


Fig. 5.16 Four clusters on the plane defined by second and third PCs.

Cluster Interpretation

Figure 5.17 shows the centroid values of the three principal components for the four clusters. Cluster sizes were 416, 625, 309, and 848 ships, respectively. ANOVA tests confirmed statistically significant differences in all principal components across these clusters ($p < 0.0001$).

The four clusters can be summarized as:

- Cluster 1: Large, environmentally friendly ships;
- Cluster 2: Small, environmentally friendly ships;
- Cluster 3: Large, non-environmentally friendly ships;
- Cluster 4: Small, non-environmentally friendly ships.

These categorizations align with existing maritime sustainability literature [187, 189].

Although the operating region index (PC3) contributed less to cluster differentiation, it showed meaningful patterns. For example, environmentally friendly small ships (Cluster 2) tend to operate exclusively within Member State jurisdictions. Additionally, there is a notable group of small, environmentally unfriendly ships (Cluster 4) operating mainly outside these jurisdictions, a relevant finding consistent with prior vessel design and operation studies [171, 172].

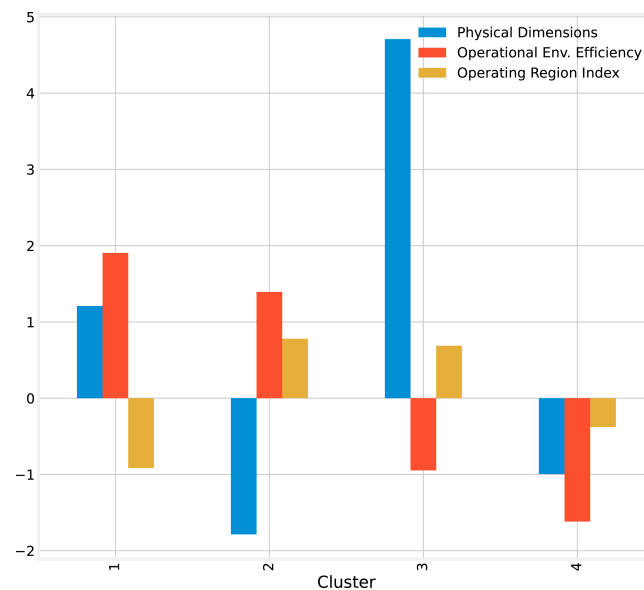


Fig. 5.17 Centroid values of the three principal components for the four k-means clusters.

Further Investigation of the Characteristics of Clusters

To deepen the understanding of the identified clusters, correlation analyses among the three PCs were conducted within each cluster. Table 5.3 presents Pearson correlation coefficients alongside corresponding p-values for all PC pairs per cluster. Despite being weak, all correlations are statistically significant at the 0.05 level.

Specifically, Physical Dimensions (PC1) and Operational Environmental Efficiency (PC2) exhibit a weak positive correlation across all clusters. This suggests that larger ships generally have a proportionally greater environmental impact. This correlation is stronger in the clusters comprising large vessels (Clusters 1 and 3) and weaker among smaller ships (Clusters 2 and 4).

Operational Environmental Efficiency (PC2) and Operating Region (PC3) show a weak negative correlation in all clusters except Cluster 2, composed of small, environmentally friendly ships. This exception may reflect that environmentally better-performing small ships tend to use cleaner fuels and operate predominantly within Member State jurisdictions. The negative correlation elsewhere may indicate that larger ships trading internationally (outside Member State jurisdictions), possibly using lower-quality fuels, possess advanced engine specifications mitigating emissions and thus exhibit relatively improved environmental efficiency.

Regarding Physical Dimensions (PC1) and Operating Region (PC3), weak but statistically significant negative correlations occur in environmentally friendly clusters

(1 and 2), whereas positive correlations exist for non-environmentally friendly clusters (3 and 4). This pattern further supports the inference that internationally operating large ships often invest in better engine technology to reduce travel costs and emissions.

Table 5.3 Pearson correlation coefficients with p-values (in parentheses) among PCs within each identified cluster.

Cluster 1: Large, environmentally friendly ships		
	Physical Dimensions (PC1)	Operational Env. Efficiency (PC2)
Operational Env. Efficiency (PC2)	0.287 (<0.001)	
Operating Region (PC3)	-0.099 (0.045)	-0.250 (<0.001)
Cluster 2: Small, environmentally friendly ships		
	Physical Dimensions (PC1)	Operational Env. Efficiency (PC2)
Operational Env. Efficiency (PC2)	0.097 (0.015)	
Operating Region (PC3)	-0.184 (<0.001)	0.250 (<0.001)
Cluster 3: Large, non-environmentally friendly ships		
	Physical Dimensions (PC1)	Operational Env. Efficiency (PC2)
Operational Env. Efficiency (PC2)	0.291 (<0.001)	
Operating Region (PC3)	0.198 (<0.001)	-0.333 (<0.001)
Cluster 4: Small, non-environmentally friendly ships		
	Physical Dimensions (PC1)	Operational Env. Efficiency (PC2)
Operational Env. Efficiency (PC2)	0.097 (0.005)	
Operating Region (PC3)	0.119 (<0.001)	-0.248 (<0.001)

PCA Validation of the Indices

To assess the validity and robustness of the PCs derived through PCA and subsequently used in CA, a validation procedure was conducted using the 62 Container ships subset of the merged dataset (see Section 5.3.2). According to Regulation (EC) No 1367/2006 of the European Parliament and Council, the main distinction between General Cargo and Container vessels relates to cargo weight and volume, making the Container ship subset a suitable independent validation group.

Table 5.4 compares the PCA loadings of the first three PCs obtained from the General Cargo reduced dataset and the Container ship data. The pairwise loading differences are also reported to quantitatively assess consistency. The results demonstrate that the PCs for Container ships closely align with those from General Cargo ships, producing similar indices in nature and behavior. This consistency indicates the indices' applicability across different ship categories. Notably, no outliers were detected within the Container ship observations, supporting dataset reliability.

Table 5.4 PCA loadings for the first three PCs, including pairwise differences.

Variable	Reduced General Cargo Dataset			Container Ship Dataset			Pairwise Differences		
	PC1	PC2	PC3	PC1	PC2	PC3	PC1	PC2	PC3
Z_1	0.101	0.480	-0.082	0.202	0.422	0.004	0.101	0.057	0.085
Z_2	-0.118	0.337	0.537	0.044	0.369	0.545	0.162	0.032	0.008
Z_3	0.172	0.317	-0.456	0.235	0.184	-0.497	0.063	0.132	0.041
Z_4	0.194	0.291	-0.455	0.207	0.199	-0.576	0.013	0.092	0.120
Z_5	0.024	0.293	0.408	0.133	0.319	0.246	0.109	0.025	0.162
Z_6	-0.161	0.398	0.102	0.148	0.396	-0.096	0.309	0.002	0.198
Z_7	-0.089	0.463	0.001	0.120	0.448	0.080	0.210	0.014	0.079
Z_8	0.382	-0.040	0.203	0.330	-0.184	0.057	0.052	0.144	0.146
Z_9	0.191	0.015	0.061	0.342	-0.073	0.125	0.151	0.088	0.065
Z_{10}	0.376	-0.020	0.073	0.365	-0.181	0.078	0.011	0.161	0.005
Z_{11}	0.385	-0.046	0.108	0.360	-0.175	0.065	0.026	0.129	0.043
Z_{12}	0.362	0.024	0.102	0.364	-0.144	0.124	0.002	0.168	0.022
Z_{13}	0.382	-0.049	0.192	0.305	-0.113	0.032	0.078	0.064	0.159
Z_{14}	0.352	0.076	-0.064	0.303	-0.128	0.050	0.049	0.204	0.115
Variance Explained	0.455	0.301	0.096	0.437	0.288	0.089	0.018	0.013	0.007

5.3.4 Conclusions

Environmental sustainability has become a central focus in maritime research, driven by the need to reduce the environmental footprint of shipping operations. Traditional methods, such as “resistance modeling,” estimate fuel consumption and emissions based on vessel resistance influenced by speed and external factors like wind and waves [165, 166]. While foundational, these models struggle with complex real-world scenarios, prompting the development of alternative approaches to improve predictive accuracy [168, 190, 191, 192, 193]. These studies generally affirm that vessel speed primarily determines fuel consumption, with weather factors playing a secondary yet significant role [167, 168].

Despite extensive research, many sustainability solutions remain theoretical, and the multiplicity of environmental initiatives within the shipping industry can create confusion and additional administrative burdens.

This chapter reviewed numerous environmental initiatives and indices currently available, including those developed by the IMO. Our framework utilizes ML extensively to form composite energy efficiency indices grounded in real ship operational data. By combining PCA and clustering, the framework produces new, interpretable energy efficiency indices. These indices are represented visually through a novel graphical tag that effectively communicates the environmental impact of ships.

While various clustering and dimensionality reduction methods could be explored in future research, PCA proves well-suited to linear transformations and variable reduction in this context.

The indices derived delineate ships into four clusters based on 14 operational and design variables, distinguishing vessels by environmental impact, physical size, and operating region. For example, small environmentally friendly ships often operate exclusively within a Member State's jurisdiction, a characteristic not shared with other clusters. Conversely, many small ships with poor environmental performance operate largely outside such jurisdictions.

This comprehensive graphical tag encapsulates CO₂ emissions adjusted for cargo capacity, directly reflecting ship size, thereby distinguishing large ships with greater environmental impact from similarly sized but cleaner vessels, as well as analogous distinctions among smaller ships.

Importantly, the current dataset offers a snapshot of ship performance. Increasing data collection frequency, such as monthly or hourly measurements, will enhance data robustness, permit identification of seasonal trends, and enable early detection of changes affecting overall fleet environmental performance. Hourly data could facilitate RTM of operational environmental efficiency (the second index), evolving towards dynamic decision-support tools that adjust emission limits and alert vessels regionally.

This evolution underscores the potential of integrating sophisticated ML indices within continuous maritime environmental monitoring systems, advancing both regulatory compliance and sustainable operational practices.

5.4 Ship Engine Model Selection

ML offers significant potential benefits for the marine industry, spanning productivity enhancements [185], improved security [205], and advanced decision support [206]. Applications include analyzing sensor data to anticipate maintenance needs [207], enabling RTM to detect safety or security risks [208], and diagnosing anomalies such as unexplained increases in oil consumption that may reflect leaks or spills [209]. These techniques support the development of preventive strategies, reduce operational downtime, and improve vessel safety. Moreover, by continuously monitoring engine and system health, M aids operators in making data-driven decisions, contributing to both productivity gains and cost savings [210]. Importantly, monitoring and analysis facilitated by ML can help ensure compliance with evolving environmental regulations, especially around ship emissions [211].

Despite these advantages, the marine sector faces challenges in effectively deploying M solutions. Access to comprehensive, high-quality onboard sensor data remains difficult [212]. Data are often fragmented across stakeholders—shipowners, opera-

tors, carriers, port agents, and regulators—complicating centralization, sharing, and standardization. Data privacy concerns, commercial sensitivities, and security considerations may limit information exchange [213, 214]. Many older vessels also lack advanced digital infrastructure, hampering RTM capability. The complex regulatory landscape, with divergent reporting standards and compliance requirements, adds further implementation barriers [215, 216]. Uncertainty over data ownership and variable connectivity at sea also hinder the seamless collection and transmission of real-time data [217]. Smaller enterprises in particular face cost constraints when investing in digital modernization. Still, recent international efforts—such as IMO mandatory reporting, satellite-based AIS tracking, and emerging IoT solutions [218, 219]—point toward a more data-accessible maritime future.

Effective application of M in this context requires collaboration between maritime subject-matter experts and data scientists to design solutions that reflect operational realities. Both unsupervised (such as clustering [16]) and supervised M methods are relevant, with supervised learning commonly used for classification or regression tasks in the industry [220].

Understanding ship types is essential for both operational and safety reasons. Ship type information reveals operational limitations, cargo capacities, and environmental risk profiles. For example, tankers transporting hazardous materials present distinct safety and regulatory challenges compared to passenger vessels, and each ship type brings unique environmental considerations [221, 222, 223, 224]. Knowledge of the vessel type aids accident prevention, rescue coordination, cargo/port management, and insurance assessment [225, 226, 227, 228]. Thus, robust collection and classification of ship-specific data is fundamental for effective environmental and operational management.

Practical experience shows that direct, reliable operational and technical data—such as engine model—are often limited or incomplete across the fleet. As a result, important details that inform design and environmental compliance may rely on indirect inference or be obscured by competitive secrecy.

This section presents a supervised M approach to classify vessel engine models using available operational and technical data. The resulting engine classification can support optimal vessel design, streamline engine selection, assist in operational strategy formation, and enable engine performance evaluation before installation. The methodology leverages multiple imputation techniques to address missing information and combines several dimensionality reduction methods with a suite of classifiers to identify the most effective predictive model.

The rest of this section is structured as follows. Subsection 5.4.1 details the methods and materials, including data collection, preprocessing, imputation, dimensionality reduction, and classification model training and evaluation. Subsection 5.4.3 presents the application data and exploratory analysis. The main findings are discussed in subsection 5.4.4, and conclusions are summarized in subsection 5.4.5.

5.4.1 Materials and Methods

The main approach for classifying a vessel’s primary engine model is rooted in supervised learning, where the model “learns” from historical examples and applies this learning to new data.

The process is structured into four primary steps, as depicted in Figure 5.18, and implemented in Python (version 3.9.7). The first step involves careful selection and preprocessing of the most relevant variables describing both operational and technical aspects of vessels. These typically include, but are not limited to, ship dimensions (length, breadth, draught), displacement, engine power, cargo capacity, voyage distance, and performance parameters such as fuel consumption and emissions.

After assembling the dataset, missing values are addressed using robust imputation strategies such as Multiple Imputation by Chained Equations (MICE), helping ensure the dataset is comprehensive and reliable for subsequent analysis.

Once the data is imputed, dimensionality reduction methods are applied to identify and retain the most informative features, facilitating improved classification performance and interpretability. Finally, various M classification algorithms are compared to determine the most accurate and robust approach for vessel engine model identification.

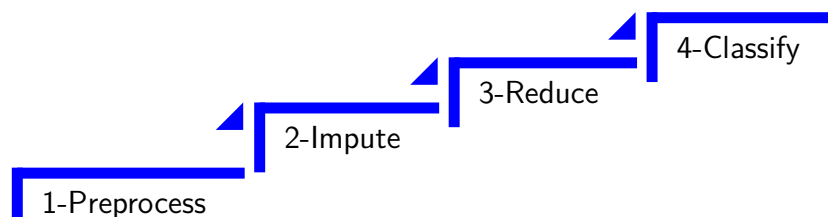


Fig. 5.18 The 4 main steps of the implementation process.

It’s important to note that the variables selected for engine model classification—ranging from physical dimensions to operational behavior—are not only critical for predicting engine type but also offer insight into key performance drivers themselves. By identifying relevant predictors, the model provides transparency about which vessel characteristics most influence engine selection, and, in turn, how these variables relate

to fuel consumption, emissions, and overall operational efficiency. This understanding is crucial for both engine validation and for informing ship design and regulatory compliance strategies.

5.4.2 Data Collection

The initial and fundamental step in any analysis or application is the collection of data. Given the variety of data sources and datasets, there is a need to create a unified and comprehensive dataset. Specifically, assume there are $i = 1, \dots, n$ data sources (or providers), each maintaining k_i , $i = 1, \dots, m$ datasets (which may be complementary or overlapping). Each dataset contains p_{ik_i} variables, where p_{ik_i} refers to the total number of variables in the k_i -th dataset of the i -th provider.

Data integration involves combining these diverse datasets into a single cohesive dataset. This process is essential when consolidating information from different sources, databases, or formats for analysis, comparison, or insight extraction.

Typically, merging data requires identifying common elements or keys to link and unify data accurately. Variables across datasets do not need to be identical, as datasets are often created for different purposes. Therefore, variable selection—keeping only the necessary variables—is crucial and can be integrated into preprocessing procedures. Continuous evaluation and updates are necessary to maintain data accuracy and relevance over time, ensuring the analyzed data remains high-quality.

Furthermore, the process of data collection and source selection should be ongoing, with regular assessments, to adapt to new data sources and evolving analytical needs.

Preprocessing and Exploratory Data Analysis

Data collection is typically the most time-intensive phase in any analytical workflow, requiring careful judgment to gather relevant and high-quality information. Ensuring data quality involves establishing rigorous collection protocols, validating sources through audits or reputation assessments, and applying best industry practices.

The accuracy of analytical outcomes depends directly on the integrity of collected data and its preprocessing. Maritime data often come unstructured and diverse, necessitating systematic preprocessing and feature exploration before predictive modeling. This stage may include feature selection to retain relevant variables, identification and treatment of missing values and duplicates, and thorough descriptive statistics.

Crucially, exploratory data analysis evaluates relationships among variables, often through correlation analyses, to inform model construction and identify potential

predictor dependencies. Understanding these inter-variable associations guides feature engineering and enhances model interpretability, ultimately improving classification or regression performance in marine operational contexts.

Data Imputation

After assessing the extent of missing values in each variable, the subsequent objective is to estimate these missing values to enrich the dataset. While deleting variables with excessive missingness is an option, it risks discarding important information and introducing bias, thereby degrading predictive model reliability [229]. Thus, statistical estimation methods for missing data imputation are preferred.

Imputation involves replacing missing entries with plausible values inferred from observed data using appropriate statistical techniques. The MICE method is employed [230]. MICE iteratively models each variable containing missing data as a function of other variables. For a variable Y with missing values, observations where Y is present inform the predictive model, which then imputes the missing entries. This iterative cycle repeats with random sampling, culminating in missing values replaced by the mean of predicted values.

MICE assumes that data are Missing At Random (MAR) or Completely At Random (MCAR) [231]. Violations of these assumptions may introduce bias, with MCAR preferred as it yields unbiased estimates, while MAR can entail negligible bias [232]. Little's test [233] can assess MCAR assumptions. Predictive models within MICE framework may include Light Gradient Boosting, Extreme Gradient Boosting, or Bayesian Ridge regression [234, 235, 236].

To evaluate MICE's accuracy and reliability, predictions on artificially masked subsets of known data are compared using Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) metrics [237, 238]. Additionally, an F-test assesses whether predictions align closely with true values by testing if intercept and slope of the population linear relationship equal zero and one, respectively, indicating unbiased prediction distribution around the ideal 45° line in scatterplots [239, 240, 241, 242, 243]. MICE-based imputation thus supports robust handling of missing maritime operational data, essential for building reliable M models in environmental performance assessment.

Dimensionality Reduction Methods

Dimensionality reduction is frequently applied before classification algorithms to eliminate redundant features, as well as noisy and irrelevant data, thereby improving the

accuracy of the learning features [244]. To identify the optimal dimensionality reduction technique suitable for our data set, various methods were evaluated. In particular, the PCA, UMAP, and t-SNE were selected. These three methods are briefly described below.

PCA

As part of the analysis, PCA is applied to the imputed data set to create new uncorrelated variables known as components, which are linear combinations of the original variables. The primary advantage of PCA is its ability to reduce dimensionality while retaining as much of the original data's variation as possible. These components may also reveal latent relationships as a result of underlying common factors. For detailed methodological explanations and applications of PCA, readers may consult [200, 245, 246, 247]. The number of components to retain can be determined using graphical techniques such as the *Scree plot* or the *Explained Variance plot*, or by applying Kaiser's criterion [248], which recommends retaining components with eigenvalues greater than one.

UMAP

UMAP was introduced recently [249] to reduce high-dimensional data into a lower-dimensional space while preserving both global and local structure. It assumes that data are uniformly distributed across a manifold suitable for projection into fewer dimensions. Unlike PCA, UMAP can capture nonlinear structures in data. It is commonly used for visualization [250] or as preprocessing before classification [251] or clustering [252]. Visualization techniques like Andrews curves have also been applied to multivariate data [253, 254].

Selecting optimal UMAP hyperparameters, such as the number of neighbors, distance metrics, and components, is challenging and typically involves experimentation. For visualization, two or three components usually suffice, but the component count can be adjusted to optimize performance for downstream tasks such as clustering or classification.

t-SNE

t-SNE maps high-dimensional data into two- or three-dimensional representations, making it valuable for visualization across diverse domains including NLP [255], geological [256, 257, 258] and health data [259, 260, 261]. It is an advancement of the

original Stochastic Neighbor Embedding (SNE) [262], reducing central point crowding and yielding superior visualizations. t-SNE excels at producing single maps that capture structure at multiple scales better than earlier methods. The key parameter to initialize in t-SNE is perplexity, influencing the effective number of nearby points considered, commonly ranging between 5 and 50 [263]. Comparative results of t-SNE with PCA and UMAP are reported in Section 5.4.4.

Classification

Classification represents the systematic assignment of data or observations into predefined categories based on shared characteristics. This process employs specific rules or algorithms derived from training data to allocate items to appropriate classes. It establishes a structured framework for organizing complex datasets, facilitating efficient pattern recognition and information retrieval. Rigorous evaluation of classification performance is critical to ensure reliability and effectiveness, serving as an essential checkpoint throughout the analytical workflow for assessing model accuracy.

Classification algorithms are formalized computational methods central to M and data analysis. These algorithms categorize data instances into predefined classes based on their attributes by applying mathematical or logical rules learned from training data. Common classification algorithms include DTs, kNN, SVMs, and ANNs. The following subsections describe the classifiers that demonstrated the highest performance in this study. Due to the wide availability of classification algorithms, selection was based on performance as detailed in Section 5.4.4.

Decision Tree Classifier

DTs are non-parametric supervised learning methods applied to both classification and regression tasks. They predict target variables by learning simple decision rules from data characteristics. DTs approximate patterns using sets of if-then-else rules, where increasing depth corresponds to more complex and precise models. DTs offer several advantages, including ease of interpretation and visualization, minimal data preparation requirements, and computational efficiency that scales logarithmically with sample size. They handle both numerical and categorical data effectively and are well-suited for multi-output problems. Their transparency makes them a white box model, allowing clear understanding of decision pathways.

However, DTs have limitations. They may become overly complex, leading to overfitting and poor generalization. Mitigation strategies include pruning, limiting minimum samples per leaf, or setting maximum tree depth. DTs can be unstable, with

minor data variations causing significant structural changes. Their predictions are not continuous, limiting suitability for extrapolation. Furthermore, finding an optimal DT is computationally challenging, often requiring heuristic approaches without guaranteeing global optimality. Class imbalance in datasets can bias models, necessitating data balancing before fitting.

Random Forest Classifier

Random Forest classifiers, implemented in the scikit-learn library, are based on the random forest algorithm [264]. This ensemble learning method constructs multiple DTs during training, each trained on a randomized data subset with random feature selection at each split. For classification, final predictions result from majority voting among trees; for regression, predictions are averaged across all trees. This approach mitigates overfitting and improves generalization. Key parameters include the number of trees in the forest (`n_estimators`), maximum tree depth (`max_depth`), and number of features evaluated per split (`max_features`). Unlike the original formulation [265], scikit-learn's implementation combines classifiers by averaging probabilistic predictions rather than voting for single classes.

Extra Trees Classifier

Extra Trees (Extremely Randomized Trees) is an ensemble method that extends DT methodology through deliberate randomness injection during tree construction. Unlike traditional DTs that select optimal split points, Extra Trees randomly selects both feature subsets and split points without evaluating optimality. This produces a forest of fully randomized trees, independent of training data output values. The algorithm balances prediction accuracy with robustness, making it particularly effective for noisy or high-dimensional data. As with DTs, data balancing is essential before fitting to avoid biased models due to class imbalances [266].

Naive Bayes Classifiers

Naive Bayes classifiers form a family of algorithms grounded in Bayes' theorem [267]. These algorithms predict class membership by maximizing posterior probability, assuming independence among variables (i.e., no correlation between variable pairs) and equal contribution of each variable to the prediction. Variants include Naive Bayes, Gaussian Naive Bayes, and Bernoulli Naive Bayes classifiers.

kNN Classifier

The kNN algorithm [268] is a non-parametric supervised learning method for classification and regression that makes no distributional assumptions. It predicts class membership by considering similarity between observations, assuming that proximate data points share class membership. Classification is based on the majority class among the kNN in the training set, determined by calculating distances in n-dimensional feature space. A related approach, the Nearest Centroid classifier [269], was also evaluated for predictive accuracy.

Other Classifiers

Linear classifiers predict class membership through linear combinations of independent variables [270]. Tested linear classifiers included SVMs, Linear SVC, Logistic Regression, Ridge Classifier, Ridge CV, SGD Classifier, Passive Aggressive Classifier [271], linear Perceptron, and Linear Discriminant Analysis.

Label Propagation Algorithms represent semi-supervised learning techniques that use labeled data points to predict labels for unlabeled instances [95]. Both Label Propagation and Label Spreading algorithms were employed in this study.

Ensemble Learning

Bagging (Bootstrap Aggregating) generates multiple training subsets via random sampling with replacement, trains independent base models on each subset, and combines predictions through majority voting (classification) or averaging (regression). Variants include Paste [272] (sampling without replacement), Bagging [273] (sampling with replacement), Random Subspaces [274] (feature subsets), and Random Patches [275] (sample and feature subsets).

Boosting combines multiple weak classifiers into a strong classifier capable of high prediction accuracy [276]. In this study, Light Gradient Boosting Machine and AdaBoost classifiers were evaluated.

Evaluation

Evaluation metrics for classifiers serve as fundamental tools for assessing M model performance in classification tasks. These metrics offer valuable insights into a model's ability to correctly categorize instances, enabling data scientists and practitioners to make informed decisions regarding model selection and application. A widely used metric is accuracy, which quantifies the proportion of correctly classified instances

relative to the total number of observations in the dataset. While accuracy provides a basic performance measure, it may not be appropriate for all scenarios, particularly when working with imbalanced datasets where one class significantly outnumbers others.

Balanced accuracy addresses the limitations of standard accuracy in imbalanced data contexts. When class distributions are heavily skewed, a classifier may achieve misleadingly high accuracy by predominantly predicting the majority class while neglecting the minority class. Balanced accuracy accounts for both sensitivity (True Positive Rate) and specificity (True Negative Rate), calculating their arithmetic mean to provide a more comprehensive assessment of model performance across all classes. This metric ensures equal weighting of minority and majority classes, making it especially valuable in maritime environmental monitoring applications where the cost of misclassifying critical conditions, such as high-emission events or non-compliant operational states, can have significant regulatory, environmental, or economic consequences. By providing a more balanced view of classification performance, this metric supports robust decision-making in real-world monitoring systems where accurate detection of all operational and environmental states is essential.

5.4.3 Application

To address missing data in the available dataset, a systematic approach combining imputation methods for handling numerical missing values and classification techniques for predicting categorical values was implemented. Imputation was selected because it preserves partial records while simultaneously filling missing data through informed estimation. This method is particularly relevant when missing data columns contribute significantly to the overall dataset. Furthermore, imputation aligns with the objective of maintaining dataset integrity and completeness, which is essential for comprehensive and objective analysis and ensures that classifiers can efficiently learn from available data across all characteristics [277].

The avoidance of deletion was implemented because it could result in dataset size reduction and potential loss of valuable information [278]. The preference for imputation over classification techniques specifically designed to handle missing information, such as those developed by [279] and [280], was driven by the intention to explore a wider range of classification algorithms. While classification techniques specialized for missing information can be effective in certain contexts, choosing imputation allows for a more extensive investigation of diverse classification algorithms that might not automatically address missing data.

In the context of maritime environmental monitoring, maintaining complete and accurate datasets through appropriate imputation methods is crucial for reliable emissions estimation, regulatory compliance assessment, and operational performance evaluation. The choice of imputation technique directly impacts the accuracy of environmental impact assessments, making it a critical consideration for developing robust monitoring and classification systems supporting sustainable maritime operations.

Data Collection

To implement the proposed methodology and evaluate classifier performance in correctly identifying ship engine models, data from two distinct providers were integrated using advanced text mining techniques to extract information from unstructured sources including emails (free text) and PDF files.

The first data provider, 27 Research PC, supplied three separate databases denoted DB1, DB2, and DB3. Each database contains ship characteristics extracted from different sources using varied methods, covering the period from January 2019 to November 2023. These data are available upon request from the authors subject to data provider permission.

The second data source, Thetis-MRV, is an open-access database where ship CO₂ emissions are monitored, reported, and verified (MRV) annually according to EU Regulation 2015/757, with data available from 2018 to 2022. Thetis-MRV represents a framework employed for climate change mitigation and emissions reduction, monitoring and reporting greenhouse gas emissions and related data. It is primarily associated with efforts to reduce emissions and combat climate change, especially in transport and industrial sectors.

Table 5.5 presents all data providers along with their respective observation counts and feature numbers.

Table 5.5 Description of each subset of the data set, based on data providers.

Source	Observations	Features	Year
DB1	141080	72	January 2019–November 2023
DB2	78065	81	January 2019–November 2023
DB3	4733	60	January 2019–November 2023
MRV1	12255	61	2018
MRV2	12399	61	2019
MRV3	12067	61	2020
MRV4	12290	61	2021
MRV5	12912	61	2022

Despite the substantial number of observations (285801 records), the databases contain duplicate entries and several sparse or irrelevant characteristics, such as previous ship names, principal operational locations, shipowner names and addresses, and contact information. Additionally, ship classification considers various fundamental variables (Length Overall, Beam, Design Draft, Gross Tonnage, Deadweight Tonnage) describing physical characteristics and capabilities. Due to interconnections among these variables, which define ship purpose, design, and operational aspects, ship type was removed from the analysis. Furthermore, additional engine-related information such as engine stroke (mm), engine builder, engine cylinders, and propeller specifications were excluded due to direct dependence on engine type. All available features related to operational conditions were utilized. Finally, through elimination of irrelevant features combined with removal of variables with insufficient observations, common features across datasets were selected for further analysis.

Consequently, merging these data sources and selecting the most informative features was essential for subsequent analysis. The Ship Identification Number (SIN) served as the key feature for data merging, providing reliable vessel identification since multiple ships may share names or vessels may be renamed during their operational life. Rather than describing all features to avoid unnecessary repetition given the large variable count, only feature sets (groups) used in further analysis are detailed in Tables 5.6 and 5.7.

Two main variable groups exist: the first consists of ship characteristics that remain constant, while the second corresponds to operational aspects. The dependent categorical variable, vessel engine model, was also included. Since engine model is not numerical, the dependent variable was converted from string to integer type.

Table 5.6 First group of variables (exterior measurements of a ship).

Name	Variable	Description
SIN	Ga_1	Ship identification number.
DWT	Ga_2	Deadweight of the ship.
Design Draft	Ga_3	Vertical distance between the waterline and the bottom of the hull.
LOA	Ga_4	The length of the ship.
Beam	Ga_5	The width of the ship at its widest point.
Depth	Ga_6	The depth measured at the middle of the length, from the top of the keel to the top of the deck beam at the side of the uppermost continuous deck.
Grain Capacity	Ga_7	The capacity of cargo spaces measured laterally to the outside of frames, and vertically from the tank tops to the top of the under weatherdeck beams, including the area contained within a ship's hatchway coamings.
Built year	Ga_8	The year of completion of ship.
Gross Tonnage	Ga_9	The volume of the ship in cubic meters below the main deck and the enclosed spaces above the main deck.
Net Tonnage	Ga_{10}	The volume of the cargo space.

Table 5.7 Second group of variables (operational).

Name	Variable	Description
Bal Speed	Gb_1	Speed where a vessel travels empty or largely empty.
Lad Speed	Gb_2	Speed where a vessel travels loaded.
Bal VLSFO	Gb_3	VLSFO ¹ consumption where a vessel travels empty or largely empty.
Lad VLSFO	Gb_4	VLSFO consumption where a vessel travels loaded.
Bal MGO	Gb_5	MDO/MGO ² consumption where a vessel travels empty or largely empty.
Lad MGO	Gb_6	MDO/MGO consumption where a vessel travels loaded.
VLSFO pi	Gb_7	VLSFO consumption during idle state.
MGO pi	Gb_8	MDO/MGO consumption during idle state.

¹ Very Low Sulfur Fuel Oil, ² Marine Diesel Oil/Marine Gas Oil.

Preprocessing and Exploratory Data Analysis

Data preprocessing and exploratory analysis are essential steps prior to constructing prediction models. In this work, the following procedure was implemented, involving appropriate handling of missing and related data. As shown in Tables 5.6 and 5.7, the dataset comprises 17 variables related to physical dimensions and environmental characteristics such as vessel consumption and speeds, plus the SIN and the target variable representing engine model, totaling 19 variables. The final variable set was derived by considering only common features with sufficient observations across all

available databases. The unified dataset resulted from a merging method (inner join) based on SIN as the key variable. Specifically, prior to implementing classification methods, a comprehensive preprocessing methodology was applied. The code for data preprocessing and subsequent analysis steps is available upon request from the authors.

Duplicate rows were removed using the variables from Tables 5.5 and 5.6, reducing observations from 285801 to 57004. Any feature vector containing even one missing value (including the main engine model variable) was deleted, resulting in 3219 observations. This dataset was later utilized to evaluate the imputation procedure. Initially, duplicate rows were dropped, but missing values remained in 17 out of 18 explanatory variables since SIN had no missing values. Missing value percentages for these variables ranged from 20% to 70%. These missing values required either deletion or imputation. In this work, to maximize information retention, an imputation approach was employed as described in the following subsection.

Beyond addressing missing values, computing correlations between variables provides initial insight into data structure. Correlations can be used as a preliminary check to verify the necessity of feature selection methods (i.e., selecting a variable subset or combining information) to reduce data space dimensionality [281]. Variables exhibiting strong correlations indicate the need for dimensionality reduction methods to achieve more concise and explanatory results. To this end, a correlation matrix was constructed. Spearman's (Sp) correlation coefficient was used for 3219 observations since none of the variables were normally distributed according to the Kolmogorov–Smirnov normality test.

The analysis revealed that variables related to physical dimensions exhibit high positive correlations ($Sp > 0.7$). Specifically, variables Ga_2 to Ga_7 , Ga_9 , and Ga_{10} showed strong positive correlations (Figure 5.19). Regarding remaining variables, correlations are less strong and do not form a wide group of highly correlated variables. Considering variables associated with environmental characteristics and vessel performance, only a few variable pairs present high positive correlation. These pairs are Gb_1 and Gb_2 , Gb_3 and Gb_4 , and Gb_5 and Gb_6 (Figure 5.19), which is unsurprising since they refer to the same measurement under two different conditions (vessel traveling empty or largely empty versus traveling loaded).

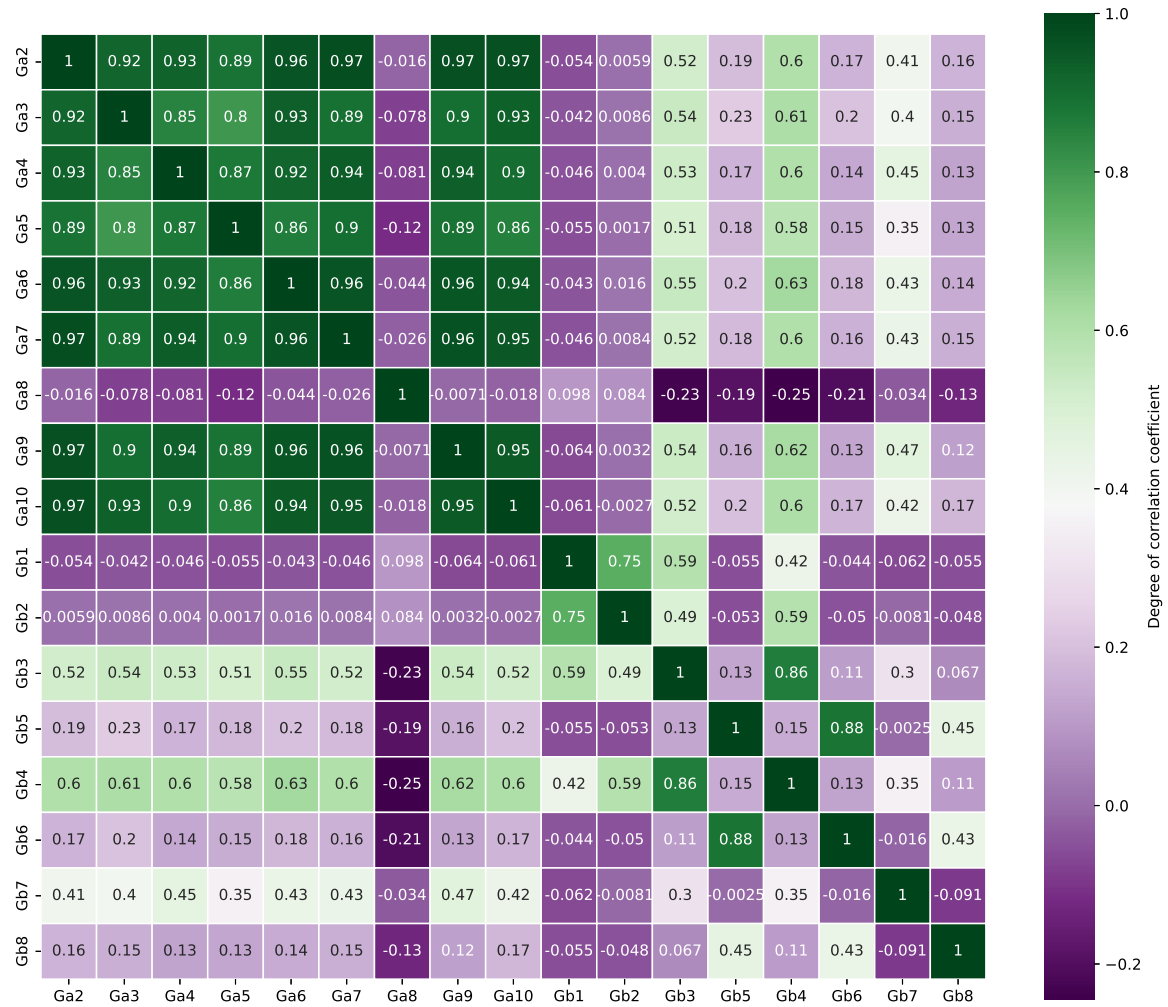


Fig. 5.19 Spearman's correlation coefficients between the independent variables.

Imputation

Consistent with the methodology outlined in Section 5.4.1, the MICE algorithm was applied to the cleaned dataset to address missing values in the explanatory variables. By implementing imputation on the 57004 observations to maximize information utilization, a final dataset of 7773 observations was obtained after removing duplicate entries and excluding any records with missing engine model information. This refinement process produced a more consolidated and representative dataset suitable for subsequent analytical procedures.

Before implementing the MICE algorithm, Little's test was conducted to verify whether data were MCAR. The p-value of the test indicated that the missing data were indeed MCAR (p-value > 0.05), meaning any missing value can be considered

unrelated to the unknown value of the variable or to other variables. The MICE algorithm was executed for a maximum of ten iterations, and three regressors were compared as estimator parameters to predict the response variable. Light Gradient Boosting [234], Extreme Gradient Boosting [235], and Bayesian Ridge regressors [236] were employed, each for a specific number of variables. The appropriate regressor was selected according to evaluation metric scores to achieve optimal imputation.

A two-step imputation strategy was utilized to capitalize on the lower proportion of missing values in the first variable set. Initially, only variables in this set were considered during the imputation process for any missing values. Subsequently, all variables from the “operational” and “exterior measurements” groups were included to impute the remaining missing values.

The comparison of evaluation metric values and F-test results are depicted in Tables 5.8–5.10. Results indicate that when the Bayesian Ridge estimator served as the estimator parameter in the MICE algorithm, the algorithm failed to generate accurate and reliable estimates for all variables (see, for example, the p-value of the F-test for the Gb_5 and Gb_6 variables in Table 5.8). Conversely, the other two approaches demonstrated superior performance by generating unbiased predictions. Among these two approaches, the most accurate MICE imputation implementation was achieved using the Extreme Gradient Boosting regressor as the estimator parameter, since the values of the three evaluation metrics were generally smaller or at least comparable for all variables relative to the corresponding values obtained using the Light Gradient Boosting estimator.

In the context of maritime environmental monitoring, accurate imputation of operational variables such as fuel consumption and vessel speeds is critical for reliable emissions estimation and regulatory compliance assessment. The selection of appropriate imputation methods directly impacts the quality of environmental performance evaluation and the validity of engine model classification, ultimately supporting more effective monitoring systems for sustainable maritime operations.

Table 5.8 Evaluating metrics of MICE imputation, considering the Bayesian Ridge estimator as a parameter for the imputation algorithm.

Variables	MSE	RMSE	MAE	F-Test <i>p</i> Values
Ga_2	13.12×10^6	3623.057	1851.556	0.063
Ga_3	0.403	0.635	0.339	0.054
Ga_4	34.446	5.869	4.282	0.152
Ga_5	1.422	1.192	0.702	0.130
Ga_6	0.567	0.753	0.494	0.075
Ga_7	19.13×10^6	4443.108	2357.83	0.086
Ga_8	13.099	3.612	2.666	0.073
Ga_9	3.74×10^6	1934.556	975.555	0.078
Ga_{10}	2.46×10^6	1570.111	832.247	0.076
Gb_1	0.425	0.652	0.470	0.143
Gb_2	0.391	0.626	0.446	0.098
Gb_3	10.664	3.265	2.310	0.127
Gb_4	27.881	5.281	2.123	0.103
Gb_5	0.015	0.122	0.054	0.004
Gb_6	0.020	0.143	0.053	0.007
Gb_7	1.168	1.108	0.508	0.057
Gb_8	0.084	0.290	0.167	0.065

Table 5.9 Evaluating metrics of MICE imputation, considering the Light Gradient Boosting estimator as a parameter for the imputation algorithm.

Variables	MSE	RMSE	MAE	F-Test <i>p</i> Values
Ga_2	1.06×10^6	1030.534	377.893	0.232
Ga_3	0.399	0.632	0.199	0.258
Ga_4	1.757	1.325	0.445	0.136
Ga_5	0.163	0.404	0.113	0.143
Ga_6	0.028	0.169	0.060	0.185
Ga_7	2.84×10^6	1685.302	535.409	0.097
Ga_8	2.248	1.499	1.034	0.137
Ga_9	6.07×10^5	779.478	261.106	0.096
Ga_{10}	1.29×10^6	1139.792	219.017	0.146
Gb_1	0.182	0.427	0.278	0.201
Gb_2	0.187	0.433	0.271	0.233
Gb_3	5.620	2.370	1.189	0.219
Gb_4	5.721	2.392	1.142	0.182
Gb_5	0.014	0.120	0.029	0.056
Gb_6	0.022	0.150	0.039	0.061
Gb_7	0.202	0.449	0.327	0.153
Gb_8	0.032	0.179	0.090	0.166

Table 5.10 Evaluating metrics of MICE imputation, considering the Extreme Gradient Boosting estimator as a parameter for the imputation algorithm.

Variables	MSE	RMSE	MAE	F-Test <i>p</i> Values
Ga_2	2.9×10^6	1711.004	265.092	0.304
Ga_3	0.288	0.536	0.166	0.279
Ga_4	1.280	1.131	0.236	0.455
Ga_5	0.137	0.370	0.076	0.226
Ga_6	0.027	0.166	0.037	0.314
Ga_7	4.60×10^6	2146.545	432.547	0.151
Ga_8	1.378	1.174	0.745	0.186
Ga_9	4.4×10^5	670.529	155.101	0.115
Ga_{10}	1.78×10^6	1337.567	168.707	0.247
Gb_1	0.192	0.438	0.258	0.293
Gb_2	0.188	0.434	0.255	0.356
Gb_3	4.262	2.064	1.038	0.346
Gb_4	2.568	1.602	0.905	0.206
Gb_5	0.012	0.111	0.024	0.078
Gb_6	0.020	0.142	0.032	0.081
Gb_7	0.174	0.417	0.295	0.254
Gb_8	0.034	0.184	0.082	0.188

Dimensionality Reduction

After completing the imputation process, PCA was performed twice using the variables from Tables 5.6 and 5.7. The first application utilized the original dataset (3219 observations), while the second used the imputed dataset (7773 observations) to compare outcomes between both implementations. Variables were standardized prior to PCA execution to ensure equal contribution to the analysis.

Although the number of variables subjected to dimensionality reduction is not particularly large, these techniques were applied for comparative purposes and to detect potential latent variables. For both PCA applications, four components were retained in the analysis since the eigenvalues of the first four principal components exceeded unity (see the left plot in Figure 5.20). For the imputed dataset, the first four PCs collectively explained 83.27% of the variability in the original data. Conversely, without imputation, the first four PCs explained a slightly lower percentage (82.47%) of the original data variability (see the right plot in Figure 5.20). This suggests that

imputation not only avoided introducing noise into the dataset but also preserved its underlying structure, justifying the use of imputed data in subsequent analyses. This is further corroborated by the similar loading values of the four PCs across both datasets.

Consequently, Table 5.11 presents PCA loadings for the first four PCs using only the imputed dataset. All large loading values for the first PC are positive, indicating a positive correlation between ships with larger physical dimensions and the first PC. For the second PC, the first four performance variables related to speed and VLSFO consumption contribute most significantly. The remaining four performance variables exhibit stronger associations with the third and fourth PCs. Additionally, in the fourth PC, variable Ga_8 (year of ship completion) demonstrates a relatively large contribution, likely reflecting evolving standards, technological advancements, and their impact on ship performance over time.

The loadings in Table 5.11 can be used to calculate PC scores for any ship. For example, the score of the first PC can be obtained as follows:

$$\begin{aligned}
 PC_1 = & 0.322 \cdot \frac{Ga_2 - 53838}{31941.41} + 0.304 \cdot \frac{Ga_3 - 11.85}{2.09} + 0.315 \cdot \frac{Ga_4 - 193.48}{26.74} + 0.310 \cdot \frac{Ga_5 - 31.22}{3.89} + \\
 & 0.311 \cdot \frac{Ga_6 - 16.75}{2.68} + 0.324 \cdot \frac{Ga_7 - 66295.6}{33194} - 0.054 \cdot \frac{Ga_8 - 2010.85}{3.54} + 0.322 \cdot \frac{Ga_9 - 30954.1}{15610.1} + \\
 & 0.322 \cdot \frac{Ga_{10} - 17930.3}{10150.8} - 0.110 \cdot \frac{Gb_1 - 13.26}{0.92} - 0.068 \cdot \frac{Gb_2 - 12.82}{0.87} + 0.261 \cdot \frac{Gb_3 - 25.2}{6.66} + \\
 & 0.262 \cdot \frac{Gb_4 - 26.09}{6.63} + 0.053 \cdot \frac{Gb_5 - 0.14}{0.16} + 0.050 \cdot \frac{Gb_6 - 0.18}{0.27} + 0.180 \cdot \frac{Gb_7 - 2.92}{0.63} + \\
 & 0.040 \cdot \frac{Gb_8 - 0.29}{0.30}.
 \end{aligned}$$

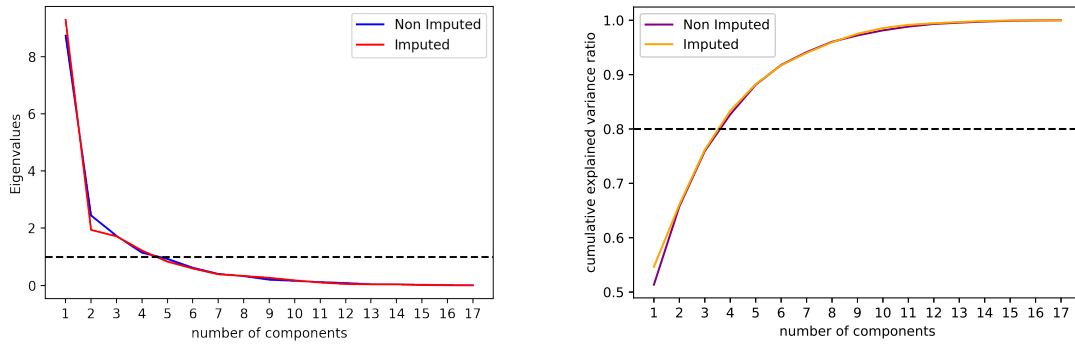


Fig. 5.20 Left plot: Scree plot of PCs for both the imputed and non-imputed datasets. Right plot: Variance explained by PCs for both the imputed and non-imputed datasets.

Table 5.11 Loadings of PCA for the imputed dataset.

Variables	PC1	PC2	PC3	PC4
Ga_2	0.322	−0.023	−0.018	−0.003
Ga_3	0.304	−0.019	0.005	0.010
Ga_4	0.315	−0.035	−0.018	0.014
Ga_5	0.310	−0.013	−0.047	−0.025
Ga_6	0.311	−0.016	−0.007	0.024
Ga_7	0.324	−0.025	−0.018	0.002
Ga_8	−0.054	0.098	−0.233	0.564
Ga_9	0.322	−0.021	−0.027	0.007
Ga_{10}	0.322	−0.027	−0.013	−0.006
Gb_1	−0.110	0.594	−0.002	0.024
Gb_2	−0.068	0.644	−0.012	0.025
Gb_3	0.261	0.304	−0.014	−0.092
Gb_4	0.262	0.339	−0.009	−0.076
Gb_5	0.053	−0.018	0.644	0.261
Gb_6	0.050	0.038	0.595	0.394
Gb_7	0.180	0.010	−0.218	0.329
Gb_8	0.040	0.078	0.350	−0.575

In addition to PCA, the t-SNE and UMAP methods were applied to compare their performance in reassessing ship engine model classification. Parameter tuning for UMAP and t-SNE, along with the impact of the three dimensionality reduction methods on classification performance, are discussed in the following section. Graphical representation was achieved by reducing data dimensionality to a two-dimensional space for both UMAP and t-SNE. For illustrative clarity, the separation of the five most frequent engine model classes is shown in Figure 5.21 using imputed data (similar patterns were observed with non-imputed data). The analytical encoding of engine models is provided in Table 5.12. All three methods (UMAP, t-SNE, and PCA) demonstrate similar separation of engine type classes, with overlapping classes also being distinguished.

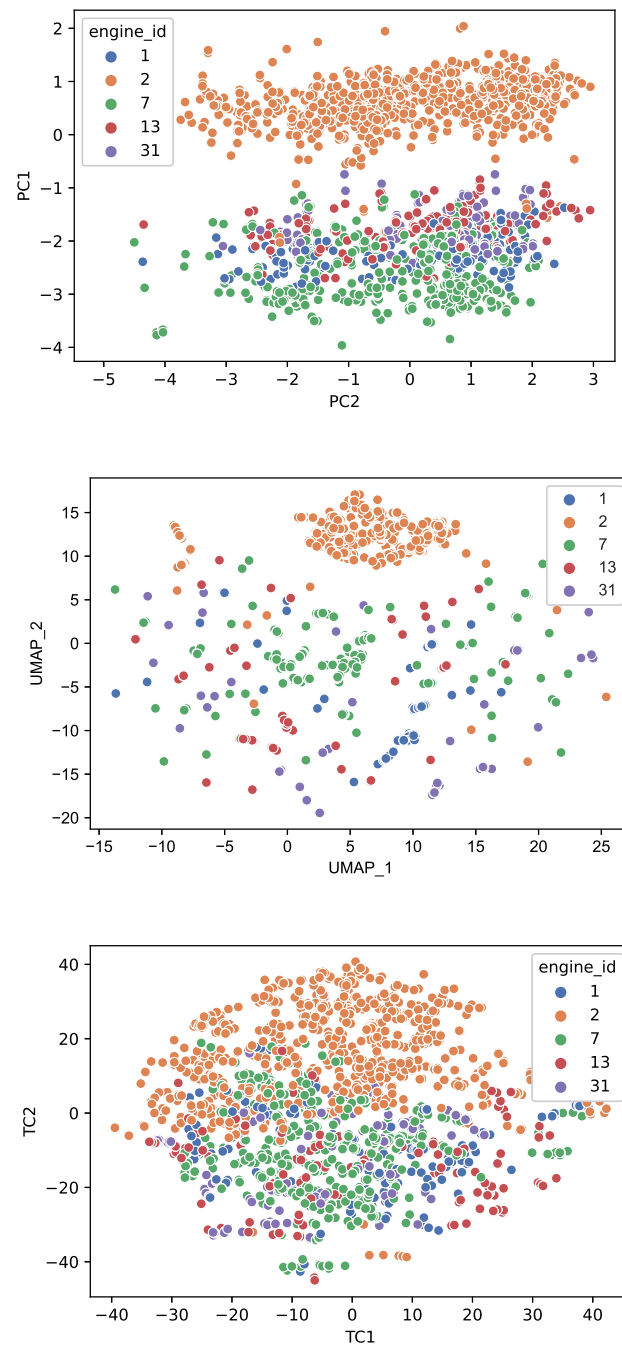


Fig. 5.21 The 5 most frequent engine types in the reduced two-dimensional space obtained by the three dimensionality reduction methods (upper left plot: PCA, upper right plot: UMAP, bottom plot: t-SNE).

Table 5.12 The final main engine models of the procedure along with their encoding and frequency in the dataset.

Encoding	Main Engine Model	Frequency	Encoding	Main Engine Model	Frequency
2	6S50MC-C	1791	38	6S50ME-B9	95
7	6S42MC	1082	49	6RTA48T	94
1	6UEC45LSE	369	9	6S60ME-C8	93
31	5S50MC-C	369	11	5S60MC-C	93
13	6S46MC-C	333	28	6RT-FLEX50	92
20	5S50ME-B9	231	6	6UEC45LSE-ECOB2	72
43	6S46ME-B8	194	3	6S60MC-C8	52
17	6S60MC-C	189	15	5RT-FL50D	52
19	6S50MC-C8	174	59	5S60MC-C8	46
30	7S50MC-C	168	21	6S42MC7	43
4	6S60MC	153	71	7S35MC	42
5	6UEC52LA	135	33	5G60ME-C9	38
23	6S50MC	133	110	5UEC45LSE	36
27	6RT-FL50	131	97	5S50MC-C8	33
10	5S60ME-C8	127	57	6RTA48T-B	32
25	6S70MC-C	127	81	6RT-FL48T	31
64	6S46MC-C8	115	32	6UEC50LSII	30
0	6S70MC	107	88	5RT-FLEX58T-B	26
12	5S50MC	107	77	6G70ME-C9	24
22	6S50ME-C8	100			

5.4.4 Classification Results

The substantial number of different engine models necessitated restricting the analysis to engines with at least 20 observations, yielding a total of 7159 observations. Table 5.12 presents the retained engine models along with their encoding and frequency in the final dataset.

Although data augmentation is frequently employed to address class imbalances, it does not universally represent the optimal solution for every classification problem. While SMOTE has been extensively used and proven beneficial in certain contexts, its application faces inherent limitations and potential adverse effects. Reference [282] demonstrates that datasets with extreme imbalances exhibit suboptimal performance even after synthetic sample generation. Additionally, reference [283] asserts that benefits of balancing techniques such as SMOTE are discernible for weak classifiers but not necessarily for robust ones. Consequently, since data augmentation was not employed, the balanced accuracy metric was used as a robust measure to mitigate the impact of

imbalanced datasets on evaluation metrics. The variable “SIN” was excluded from the classification process, as it clearly did not contribute relevant features for the analysis. A comprehensive evaluation of several classification methods for correctly identifying ship engine type under different dimensionality reduction methods is presented in this section. Specifically, 25 classifiers in total were compared with respect to their balanced accuracy performance under three dimensionality reduction methods. Each dimensionality reduction method was tested under different parameter configurations.

The PCA method was systematically evaluated by considering utilization of the first two, three, and four PCs. Additionally, UMAP was employed for dimensional reduction with two and three components, investigating various neighbor values (specifically 10, 30, and 123), leading to six different combinations. Furthermore, t-SNE was applied to achieve dimensionality reduction, exploring configurations with two and three components and perplexity values of 10, 30, and 123. Consequently, six parameter combinations of the t-SNE method were evaluated. This comprehensive comparative analysis aimed to determine the efficacy of these techniques under various parameter settings.

The dataset was initially partitioned into training and test sets using the “`train_test_split`” function from the “`model_selection`” module provided by the scikit-learn library. Specifically, 80% of the data were allocated for training purposes, while the remaining 20% were reserved for evaluating model performance. This division ensures that the model is trained on a substantial data portion and then tested on an independent subset to assess generalization capabilities. Moreover, a robust k-fold cross validator was also used to systematically evaluate model performance by dividing the data into five consecutive folds.

The candidate classification methods were all applied through the “`LazyClassifier`” function from the “`lazypredict.Supervised`” library in Python (version 3.9.7). The “`lazypredict`” library was initially adopted to facilitate comparison of various M models using default hyperparameters for scikit-learn classifiers. Furthermore, for the optimal classifier (ExtraTreesClassifier), additional evaluation across different values of the number of trees (`n_estimators` = [10, 50, 100, 150, 500]) pertaining to both imputation techniques and dimensional reduction methods was performed. Additionally, all classifiers were compared based on their training process duration.

Table 5.13 presents the optimal performance results corresponding to each specific feature selection method (PCA, UMAP, and t-SNE) using k-fold cross-validation, having compared the various parameters tested for each technique (namely, the number of PCs, number of neighbors, and perplexity). The dimensionality reduction methods

presented are PCA with four components (PCA-4), t-SNE with perplexity equal to 10 and three components (Tsne10-3), and UMAP with 10 neighbors and three components (Umap10-3). The largest value(s) for the balanced accuracy metric is marked in bold. For illustration purposes, the balanced accuracy values for the classifiers are also depicted in Figure 5.22 for the imputed data case.

In the context of maritime environmental monitoring, accurate engine model classification enables validation of reported technical specifications, supporting emissions estimation accuracy and regulatory compliance assessment. The high classification performance achieved demonstrates that operational and physical characteristics provide sufficient information to reliably identify engine models, which is essential for monitoring systems designed to detect anomalies or verify manufacturer-reported fuel consumption and emission characteristics under real operating conditions.

The optimal performance results presented in Table 5.13 were derived from the results shown in Tables 5.22 through 5.26 presented at the end of this chapter. Specifically, Table 5.22 demonstrates that four-component PCA paired with ExtraTreeClassifier outperforms all other combinations of classifiers and PCs. Among combinations of t-SNE with two components, t-SNE with a perplexity of 10 paired with ExtraTreeClassifier appears to yield the most favorable outcomes based on observed results (Table 5.24).

Table 5.13 presents the optimal performance results corresponding to each specific feature selection method (PCA, UMAP, and t-SNE), having compared the various parameters tested for each technique (namely, the number of PCs, number of neighbors, and perplexity). The dimensionality reduction methods presented are PCA with four components (PCA-4), t-SNE with perplexity equal to 10 and two components (Tsne10-2), and UMAP with 10 neighbors and two components (Umap10-2). The largest value(s) for balanced accuracy metric is marked in bold.

Regarding the combination of t-SNE with two components, t-SNE with a perplexity of 10 paired with ExtraTreeClassifier again yields optimal performance (Table 5.19). However, comparatively, t-SNE with three components and a perplexity of 10 exhibits better performance than t-SNE with two components and a perplexity of 10, specifically 90.20% versus 89.21

In the context of imputed data analysis, ExtraTreeClassifier emerges as the best classifier compared to others, exhibiting exceptionally strong performance with 95.07% accuracy. Notably, as can be derived from the above, when evaluated across various dimensionality reduction methods, ExtraTreesClassifier consistently demonstrates exceptional performance, outperforming other classifiers. While these findings highlight the consistent proficiency of ExtraTreesClassifier across diverse dimensionality reduction

Table 5.13 Performance of evaluation metrics (balanced accuracy) for imputation data and the top performance for each of the three dimensionality reduction methods using k-fold cross validation. The time taken for each algorithm is also included.

Classification	Algorithm	Balanced Accuracy			Time Taken (seconds)		
	ExtraTreesClassifier	0.9507	0.8709	0.9020	0.8364	0.3544	0.4136
	RandomForestClassifier	0.9502	0.8577	0.8931	0.8125	0.5524	1.0609
	BaggingClassifier	0.9408	0.8319	0.8700	0.7875	0.1792	0.1886
	DecisionTreeClassifier	0.9340	0.8267	0.8859	0.7928	0.0299	0.0344
	ExtraTreeClassifier	0.9034	0.8240	0.8596	0.7730	0.0106	0.0100
	LabelPropagation	0.8848	0.5514	0.6281	0.4010	1.0217	0.7152
	LabelSpreading	0.8845	0.5311	0.6157	0.3954	1.6083	1.3164
	KNeighborsClassifier	0.7034	0.5683	0.7082	0.6526	0.1459	0.0377
	LGBMClassifier	0.5920	0.5301	0.5525	0.7635	1.4434	1.5696
	LinearDiscriminantAnalysis	0.5778	0.2008	0.1147	0.1032	0.0207	0.0113
	GaussianNB	0.5775	0.3126	0.1871	0.1583	0.0146	0.0112
	LogisticRegression	0.5093	0.2023	0.1378	0.1041	0.8430	0.7778
	SVC	0.4757	0.2498	0.3073	0.2490	1.0982	1.1509
	LinearSVC	0.4678	0.0917	0.0845	0.0725	1.4458	2.1846
	NearestCentroid	0.4491	0.2534	0.2421	0.2129	0.0107	0.0086
	CalibratedClassifierCV	0.4146	0.0812	0.0847	0.0698	5.6612	8.6005
	SGDClassifier	0.3895	0.1144	0.0899	0.0881	0.3062	0.1587
	QuadraticDiscriminantAnalysis	0.3193	0.4012	0.2680	0.2310	0.0259	0.0116
	Perceptron	0.3112	0.0869	0.0893	0.0788	0.1129	0.0651
	PassiveAggressiveClassifier	0.2612	0.0777	0.0835	0.0832	0.1347	0.0696
	BernoulliNB	0.2585	0.0503	0.0451	0.0459	0.0118	0.0093
	RidgeClassifier	0.1477	0.0481	0.0438	0.0619	0.0144	0.0122
	RidgeClassifierCV	0.1477	0.0481	0.0437	0.0615	0.0281	0.0212
	AdaBoostClassifier	0.1263	0.0865	0.0680	0.0714	0.4822	0.475
	DummyClassifier	0.0256	0.0256	0.0256	0.0256	0.0078	0.0060
						0.0078	0.0064
							0.0059

techniques, it is worth noting that t-SNE stands out as the top-performing method, displaying the highest accuracy in all evaluated scenarios.

Additionally, the best-performing results obtained using “`train_test_split`” are reported, along with the results summarized in Tables 5.13–5.21. Specifically, Table 5.17 shows that four-component PCA paired with `ExtraTreeClassifier` outperforms all other combinations of classifiers and PCs. Among combinations of t-SNE with two components, t-SNE with a perplexity of 10 paired with `ExtraTreeClassifier` appears to yield the most favorable outcomes based on observed results (Table 5.18).

Based on these results, it is concluded that DT logic-based classifiers (specifically, `ExtraTreesClassifier`, `DecisionTreeClassifier`, `RandomForestClassifier`, `BaggingClassifier`, and `ExtraTreeClassifier`) perform optimally and successfully classify engine model type with great accuracy (up to 96%). In addition to the most accurate results obtained for imputed data, the reduced data achieved satisfactory outcomes in dimension reduction tests, even with only two components instead of the original 18 variables. Although various explanations exist for why these algorithms outperform others, the most common are highlighted below.

DTs adapt well to data characteristics. They create complex decision boundaries when necessary and effectively capture non-linear relationships in data structure. Other classifiers, such as linear models, may struggle when underlying patterns are not linear. Furthermore, DTs inherently perform feature selection by ranking feature importance. That is, they focus on the most important features for decision making, which provides a significant advantage when dealing with high-dimensional data.

They also serve as base models for ensemble methods, such as random forests and gradient boosting. Ensemble methods combine multiple DTs to reduce overfitting and improve prediction accuracy. This may lead to superior performance compared to standalone classifiers. Additionally, DTs can handle categorical data without requiring one-hot encoding or other preprocessing steps, simplifying the modeling process and potentially leading to better results in situations where combinations of categorical and numerical features exist.

Another advantage is that DTs are relatively robust to outliers and missing data, which are common problems in real-world data sources. They can handle these situations gracefully, potentially leading to better overall performance.

It is important to note that relative classifier performance and feature selection technique effectiveness can vary depending on the dataset and specific problem. While DT-based classifiers have these advantages, cases exist where other classifier types, such

as SVMs, ANNs, or kNNs, may be more appropriate. Classifier and dimensionality reduction method choice should be based on data characteristics and M task goals.

Table 5.14 shows balanced accuracy results from ExtraTreesClassifier over several values of the “n_estimators” hyperparameter. Columns correspond to different configurations of the “n_estimators” parameter, while rows denote imputation performance and different dimensional methods. The dataset is methodically divided into five consecutive folds using k-fold cross-validation. The final performance metric is derived by averaging scores obtained across five k-fold cross-validation iterations.

In the context of maritime environmental monitoring, the high classification accuracy achieved demonstrates that operational and physical vessel characteristics provide sufficient information to reliably identify engine models. This capability is essential for monitoring systems designed to validate

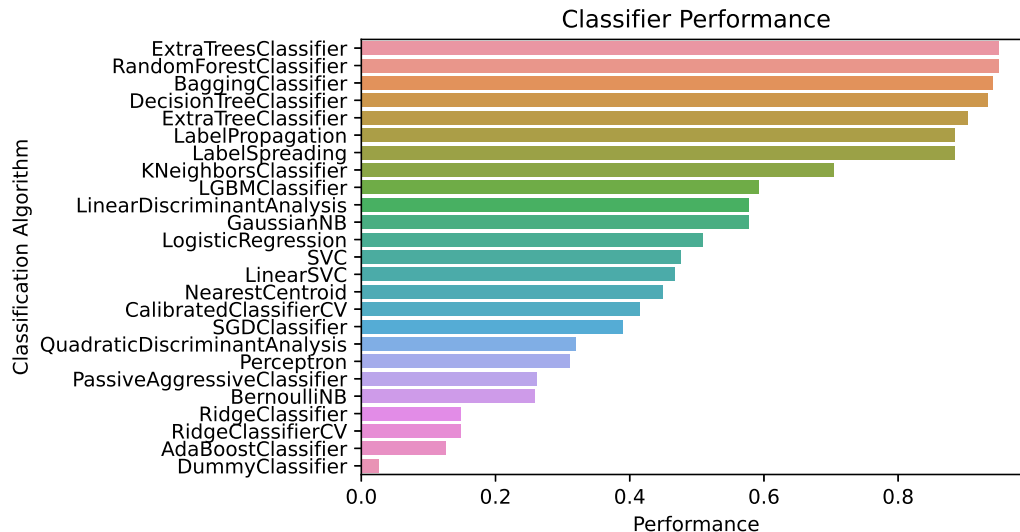


Fig. 5.22 Performance of evaluation metrics (balanced accuracy) for imputed data using k-fold cross validation.

While the classifier results are deemed satisfactory, it is important to examine the source of minor classification errors. When considering Table 5.15 based on results from the top classifier (ExtraTreesClassifier) along with the illustration in Figure 5.21, a pattern can be observed among certain encoded engine types (the five most frequent). Combining the visualization, which utilized dimensionality reduction techniques, and the confusion matrix based on the imputed dataset, it is evident that misclassifications occur toward specific engine types in all instances. This can be attributed to the presence of similar operational and physical characteristics accompanying these engine models.

Table 5.14 The average balanced accuracy scores from five iterations of the k-fold cross-validation technique of the ExtraTreesClassifier across different n_estimators values (for imputed and non-imputed data as well as for multiple dimensionality reduction methods).

n	10	50	100	150	500
Imput	0.9491	0.9537	0.9479	0.9513	0.9541
Umap-2-10	0.8089	0.8267	0.8205	0.8189	0.8273
Umap-2-30	0.7309	0.7454	0.7468	0.7555	0.7523
Umap-2-123	0.5896	0.6174	0.6142	0.6144	0.6186
Umap-3-10	0.8251	0.8224	0.8322	0.8218	0.8357
Umap-3-30	0.7672	0.7886	0.8015	0.7911	0.7963
Umap-3-123	0.6823	0.7089	0.7200	0.7076	0.7177
PCA-2	0.7474	0.7673	0.7701	0.7753	0.7693
PCA-3	0.8228	0.8285	0.8280	0.8354	0.8282
PCA-4	0.8522	0.8594	0.8630	0.8614	0.8667
tsne-2-10	0.8880	0.9048	0.8954	0.8947	0.8941
tsne-2-30	0.8820	0.8910	0.8934	0.8910	0.8996
tsne-2-123	0.8786	0.8725	0.8801	0.8805	0.8813
tsne-3-10	0.8961	0.8913	0.8861	0.9029	0.9005
tsne-3-30	0.8919	0.8954	0.8927	0.8870	0.8899
tsne-3-123	0.8753	0.8840	0.8777	0.8841	0.8830

In the context of maritime environmental monitoring, these misclassifications predominantly occur between engine models that share similar power output ranges, fuel consumption profiles, and vessel size compatibility. This observation suggests that engines with comparable technical specifications and performance characteristics may exhibit similar operational signatures, making them challenging to distinguish solely based on operational data. Understanding these patterns is valuable for refining monitoring systems and recognizing when additional verification of manufacturer-reported specifications may be necessary to ensure accurate emissions estimation and regulatory compliance assessment.

Table 5.15 Confusion matrix of the 5 most frequent engine models (imputed data) and $n_estimator = 50$.

Index	1	2	7	13	31
1	68	0	1	1	0
2	0	358	0	0	0
7	5	0	212	0	0
13	0	1	0	65	0
31	0	0	0	0	71

5.4.5 Conclusions

The capability to correctly classify a ship's engine type using predominantly available data provides an important advantage in the highly competitive shipping industry. To achieve this, a thorough examination was conducted to identify the optimal dimensionality reduction method and classifier. Identification of the best combination was made through detailed comparison and evaluation of their performance.

This classification procedure enables important, innovative, and valuable conclusions for the marine industry. Ship industries could reduce operating costs if the type of main engine of their vessels is known. Consequently, given the physical dimensions of ships and their operational behavior, shipowners can select and deploy the appropriate engine model for their fleet depending on their strategic plan and operational schedule.

Furthermore, if shipping companies are aware of which main engine models cause higher fuel consumption, they can decide which engine types to use with the ultimate aim of reducing emissions from ships, thus sailing legally under rigorous international and environmental regulations without any burdens.

The optimal main engine model used in most cases may motivate shipping companies to purchase such engine models because these may improve the durability and sustainability that ships possess in terms of both voyage time and operating costs. This reinforced sustainability could reduce the occurrence of malfunctioning vessels by avoiding unexpected and undesirable accidents in the future, which may result in serious damage and loss of life.

Another inference of this study is that marine industries, provided they have a large amount of data at their disposal, could make safe and reliable decisions for their strategic plan with a smaller subset of data referring mainly to physical and environmental attributes.

A more detailed presentation of the advantages of this method over traditional ones could include, for instance, faster engine selection, developing optimal strategies for a new vessel's use, or even developing digital twin applications or simulators, allowing evaluation of ship performance before engine placement. The last will result in operating cost reduction. The above-mentioned digital twins can be an asset in performance monitoring and predictive maintenance.

Considering faster engine selection, the proposed model can be used during the ship design phase. That is, a quick approximate assessment and identification of the optimum marine engine could be provided. It is noteworthy that hundreds of different machine types exist in the market; thus, the capability provided by the proposed methodology for quick screening and selection of the optimal engine type is of great value.

Selecting the optimal engine for the vessel, which emerges naturally by using the proposed methodology, can be integrated into the field of developing optimal use strategies. This can be achieved by considering the physical dimensions of the vessel and strategically determining the intended use of the ship as well as the desired operational behavior. For instance, the consumption rates and speeds all arise from its shipowner's strategy.

As already mentioned, applications of digital twins could be developed as well. These digital models are developed using operational behavior data, historical information, and advanced analytics. They mirror the physical counterpart's characteristics, behavior, and performance in a virtual environment. Thus, the shipbuilding industry can reduce operating costs by knowing which type of main engine is optimum for a vessel.

Furthermore, the proposed methodology offers a tool for reducing harmful emissions from ships and their wider use in accordance with strict international environmental standards without any restrictions. Moreover, such classification procedures may lead to the identification of important innovative and valuable findings for the maritime industry.

From a maritime environmental monitoring perspective, the proposed methodology provides critical capabilities for validating manufacturer-reported engine specifications against actual operational data. By accurately identifying engine models from operational and physical characteristics, monitoring systems can detect discrepancies between expected and observed fuel consumption and emissions patterns. This verification capability is essential for ensuring regulatory compliance, detecting potential data misreporting, and supporting accurate emissions inventories. The high classification accuracy achieved (95.07%) demonstrates that operational monitoring data contains

sufficient information to reliably identify engine types, making this approach valuable for both RTM and historical data validation in environmental compliance assessment systems.

The main limitation of this study is that there was insufficient information on all of the 123 available engine models of the ships. Despite exploiting the entirety of available features and data, encompassing both operational and design data, information on the operating conditions of ships or correspondingly whether they pertain to personalized features of their design was lacking. Additionally, a considerable number of missing values existed in variables pertaining to the operational efficiency of ships.

The proposed methodology reflects the actual data used to train the model in this research, representing the current reality in ships' design and engine selection. However, the current reality, as depicted in the data, encompasses both best practices and less or non-optimal engine choices. Naturally, it is expected that good or best practices have been applied to the majority of vessels at our disposal. Through the extensive dataset available, the proposed method identifies the underlying data structure and selects the prevailing strategy. Therefore, it validates the best strategies on one hand, while on the other, it provides a quick and straightforward way for the universal application of best strategies during the design phase.

In conclusion, whilst the estimated missing values were accurate, the outcomes would have been more reliable and robust if more observations existed for this group of variables. Consequently, it would be wise and prudent to gather as much data as possible regarding the operational efficiency of ships in future studies, if the challenging circumstances and competition permit this.

5.5 Conclusion

Environmental sustainability in maritime operations has become increasingly critical as the shipping industry faces mounting pressure to reduce its ecological footprint. This research addresses fundamental challenges in maritime environmental assessment and operational optimization through integrated M frameworks that leverage real-world operational data. While traditional resistance modeling approaches have provided foundational methods for estimating fuel consumption and emissions, their limitations in capturing the complexity of actual shipping operations have necessitated the development of more sophisticated, data-driven methodologies.

This work contributes two complementary advances to maritime sustainability. First, it develops composite energy efficiency indices through the application of PCA and

Table 5.16 *Cont.* Performance of evaluation metrics (balanced accuracy) and time taken for each algorithm.

Classification Algorithm	Balanced Accuracy				Time Taken (seconds)			
	Input	PCA-4	Tsne10-2	Umap10-2	Input	PCA-4	Tsne10-2	Umap10-2
LabelSpreading	0.9159	0.5372	0.3315	0.3295	1.4094	0.9950	0.9384	0.9421
KNeighborsClassifier	0.6964	0.5847	0.6946	0.6546	0.1481	0.0439	0.0379	0.0375
GaussianNB	0.5676	0.2970	0.1435	0.0846	0.0189	0.0189	0.0180	0.0210
LinearDiscriminantAnalysis	0.5548	0.1928	0.0642	0.0592	0.0376	0.0140	0.0100	0.0100
LogisticRegression	0.5190	0.2042	0.0746	0.0589	0.7321	0.7281	0.7288	0.7118
SVC	0.4608	0.2677	0.2216	0.1628	1.3545	1.4457	1.5451	1.8055
LinearSVC	0.4554	0.0956	0.0639	0.0577	2.0496	3.0429	0.5207	0.7550
NearestCentroid	0.4481	0.2532	0.1928	0.1596	0.0110	0.0090	0.0090	0.0090
CalibratedClassifierCV	0.4287	0.0929	0.0562	0.0524	8.3219	11.5316	2.5174	3.3163
SGDClassifier	0.3982	0.1119	0.0805	0.0623	0.3361	0.1725	0.1272	0.1227
Perceptron	0.3587	0.0920	0.0396	0.0211	0.1287	0.0768	0.0478	0.0598
PassiveAggressiveClassifier	0.3157	0.0695	0.0704	0.0348	0.1546	0.0678	0.0618	0.0598
BernoulliNB	0.2333	0.0504	0.0570	0.0386	0.0352	0.0121	0.0110	0.0119
RidgeClassifier	0.1438	0.0497	0.0390	0.0321	0.0249	0.0199	0.0190	0.0179
RidgeClassifierCV	0.1438	0.0497	0.0390	0.0321	0.0284	0.0199	0.0219	0.0189
AdaBoostClassifier	0.1266	0.0766	0.0671	0.0389	0.4466	0.4468	0.3829	0.3832
DummyClassifier	0.0256	0.0256	0.0256	0.0256	0.0090	0.0090	0.0110	0.0070
QuadraticDiscriminantAnalysis	0.0256	0.3961	0.1391	0.1194	0.0199	0.0130	0.0189	0.0130

Table 5.17 Performance of evaluation metric (balanced accuracy) for imputation data using PCA method for different components using “train_test_split”, along with the time taken for training each algorithm.

Classification Algorithm	Balanced Accuracy			Time Taken (seconds)		
	PCA-2	PCA-3	PCA-4	PCA-2	PCA-3	PCA-4
ExtraTreesClassifier	0.7914	0.8601	0.8968	0.3680	0.4675	0.4599
LGBMClassifier	0.7213	0.8557	0.8862	1.5523	2.0348	1.5234
RandomForestClassifier	0.7613	0.8610	0.8942	0.6910	0.7451	1.1152
DecisionTreeClassifier	0.7509	0.8247	0.8699	0.0250	0.0299	0.0397
BaggingClassifier	0.7297	0.8336	0.8705	0.1192	0.1630	0.1985
ExtraTreeClassifier	0.7110	0.7700	0.8390	0.0100	0.0110	0.0110
LabelPropagation	0.2029	0.3442	0.5606	0.6273	0.6662	0.6717
LabelSpreading	0.1980	0.3391	0.5372	1.0045	1.0298	1.1671
KNeighborsClassifier	0.4938	0.5456	0.5847	0.0379	0.0409	0.0439
GaussianNB	0.1352	0.2187	0.2970	0.0209	0.0259	0.0222
LinearDiscriminantAnalysis	0.1393	0.1564	0.1928	0.0100	0.0110	0.0140
LogisticRegression	0.1171	0.1252	0.2042	0.7938	0.8086	0.7721
SVC	0.1445	0.1746	0.2677	1.5649	1.5430	1.4709
LinearSVC	0.0604	0.0684	0.0956	0.5096	2.7077	3.0354
NearestCentroid	0.1761	0.1765	0.2532	0.0090	0.0090	0.0110
CalibratedClassifierCV	0.0562	0.0616	0.0929	2.5534	10.7399	11.6460
SGDClassifier	0.0679	0.0776	0.1119	0.1057	0.1411	0.1725
Perceptron	0.0388	0.0723	0.0920	0.0529	0.0598	0.0738
PassiveAggressiveClassifier	0.0568	0.0617	0.0695	0.0549	0.0588	0.0819
BernoulliNB	0.0504	0.0504	0.0504	0.0110	0.0119	0.0109
RidgeClassifier	0.0502	0.0497	0.0497	0.0180	0.0199	0.0169
RidgeClassifierCV	0.0501	0.0497	0.0497	0.0379	0.0229	0.0220
AdaBoostClassifier	0.0766	0.0766	0.0766	0.4220	0.4561	0.4828
DummyClassifier	0.0256	0.0256	0.0256	0.0070	0.0080	0.0080
QuadraticDiscriminantAnalysis	0.1575	0.2326	0.3961	0.0150	0.0131	0.0130

Table 5.18 Performance of evaluation metrics (balanced accuracy) for imputation data using t-SNE with 2 components and different perplexity values using “train_test_split”, along with the time taken for training each algorithm.

Classification Algorithm	Balanced Accuracy			Time Taken (seconds)		
	Tsne10-2	Tsne30-2	Tsne123-2	Tsne10-2	Tsne30-2	Tsne123-2
ExtraTreesClassifier	0.9206	0.9173	0.9132	0.3680	0.3640	0.3705
LGBMClassifier	0.9105	0.8904	0.8719	1.5523	1.9767	1.8871
RandomForestClassifier	0.9136	0.9139	0.9074	0.6910	0.6855	0.7211
DecisionTreeClassifier	0.9064	0.8964	0.8991	0.0250	0.0239	0.0239
BaggingClassifier	0.8951	0.8887	0.8839	0.1192	0.1141	0.1219
ExtraTreeClassifier	0.9049	0.8954	0.8948	0.0100	0.0110	0.0110
LabelPropagation	0.3408	0.2747	0.2292	0.6273	0.6551	0.6260
LabelSpreading	0.3315	0.2724	0.2240	1.0045	1.0087	1.0044
KNeighborsClassifier	0.6946	0.6890	0.6573	0.0379	0.0389	0.0389
GaussianNB	0.1435	0.1832	0.1400	0.0209	0.0209	0.0219
LinearDiscriminantAnalysis	0.0642	0.1241	0.1320	0.0100	0.0120	0.0100
LogisticRegression	0.0746	0.1270	0.1328	0.7938	0.7783	0.7849
SVC	0.2216	0.2309	0.1747	1.5649	1.4505	1.3898
LinearSVC	0.0639	0.0458	0.1009	0.5096	0.4235	0.4019
NearestCentroid	0.1928	0.2227	0.1823	0.0090	0.0115	0.0100
CalibratedClassifierCV	0.0562	0.0592	0.0856	2.5534	1.9490	2.1004
SGDClassifier	0.0805	0.0928	0.1019	0.1057	0.1097	0.1177
Perceptron	0.0396	0.0491	0.0927	0.0529	0.0520	0.0519
PassiveAggressiveClassifier	0.0704	0.0605	0.1028	0.0549	0.0610	0.0588
BernoulliNB	0.0570	0.0442	0.0692	0.0110	0.0110	0.0113
RidgeClassifier	0.0390	0.0409	0.0538	0.0180	0.0189	0.0199
RidgeClassifierCV	0.0390	0.0409	0.0538	0.0379	0.0232	0.0229
AdaBoostClassifier	0.0671	0.0634	0.0731	0.4220	0.4099	0.4118
DummyClassifier	0.0256	0.0256	0.0256	0.0070	0.0070	0.0070
QuadraticDiscriminantAnalysis	0.1391	0.2037	0.2103	0.0150	0.0130	0.0120

Table 5.19 Performance of evaluation metrics (balanced accuracy) for imputation data using t-SNE with 3 components and different perplexity values using “train_test_split”, along with the training time for each algorithm.

Classification Algorithm	Balanced Accuracy			Time Taken (seconds)		
	Tsne10-3	Tsne30-3	Tsne123-3	Tsne10-3	Tsne30-3	Tsne123-3
ExtraTreesClassifier	0.9181	0.9146	0.9169	0.3680	0.3612	0.3671
LGBMClassifier	0.0427	0.9104	0.0453	1.5523	2.2633	0.9634
RandomForestClassifier	0.9171	0.9135	0.9141	0.6910	0.7550	0.7540
DecisionTreeClassifier	0.9091	0.9054	0.8967	0.0250	0.0321	0.0320
BaggingClassifier	0.8912	0.8883	0.8852	0.1192	0.1725	0.1649
ExtraTreeClassifier	0.8884	0.8951	0.8978	0.0100	0.0130	0.0100
LabelPropagation	0.6550	0.5999	0.5336	0.6273	0.6228	0.6521
LabelSpreading	0.6397	0.5715	0.5151	1.0045	0.9969	1.0462
KNeighborsClassifier	0.7047	0.6905	0.6748	0.0379	0.0399	0.0399
GaussianNB	0.1815	0.2226	0.2120	0.0209	0.0219	0.0228
LinearDiscriminantAnalysis	0.1132	0.1791	0.1520	0.0100	0.0120	0.0129
LogisticRegression	0.1398	0.1891	0.1622	0.7938	0.9065	0.9106
SVC	0.3146	0.2956	0.2561	1.5649	1.3408	1.3334
LinearSVC	0.0925	0.1385	0.1255	0.5096	0.4877	0.6084
NearestCentroid	0.2424	0.2867	0.2457	0.0090	0.0100	0.0110
CalibratedClassifierCV	0.0866	0.1378	0.1429	2.5534	2.3304	2.5602
SGDClassifier	0.0920	0.1101	0.0887	0.1057	0.1336	0.1307
Perceptron	0.0858	0.0947	0.0609	0.0529	0.0648	0.0598
PassiveAggressiveClassifier	0.0911	0.1228	0.1137	0.0549	0.0648	0.0598
BernoulliNB	0.0426	0.0607	0.0774	0.0110	0.0120	0.0120
RidgeClassifier	0.0437	0.0601	0.0499	0.0180	0.0189	0.0209
RidgeClassifierCV	0.0437	0.0601	0.0499	0.0379	0.0219	0.0229
AdaBoostClassifier	0.0574	0.0551	0.0787	0.4220	0.4538	0.4643
DummyClassifier	0.0256	0.0256	0.0256	0.0070	0.0090	0.0079
QuadraticDiscriminantAnalysis	0.2661	0.2965	0.2812	0.0150	0.0130	0.0160

Table 5.20 Performance of evaluation metric (balanced accuracy) for imputation data using UMAP with 2 components and different near-neighbor values using “train_test_split”, along with the time taken for training each algorithm.

Classification Algorithm	Balanced Accuracy			Time Taken (seconds)		
	UMAP10-2	UMAP30-2	UMAP123-2	UMAP10-2	UMAP30-2	UMAP123-2
ExtraTreesClassifier	0.8537	0.7868	0.6164	0.3680	0.4836	0.5185
LGBMClassifier	0.7620	0.0534	0.5915	1.5523	0.8876	2.0281
RandomForestClassifier	0.8416	0.7250	0.6037	0.6910	0.7556	0.7909
DecisionTreeClassifier	0.8148	0.7153	0.5816	0.0250	0.0260	0.0269
BaggingClassifier	0.8093	0.7071	0.6060	0.1192	0.1267	0.1297
ExtraTreeClassifier	0.7948	0.6567	0.5167	0.0100	0.0130	0.0120
LabelPropagation	0.3324	0.2198	0.1450	0.6273	0.6445	0.6467
LabelSpreading	0.3295	0.2127	0.1436	1.0045	1.0696	0.9904
KNeighborsClassifier	0.6546	0.5961	0.5631	0.0379	0.0379	0.0379
GaussianNB	0.0846	0.1651	0.1901	0.0209	0.0279	0.0315
LinearDiscriminantAnalysis	0.0592	0.1074	0.1084	0.0100	0.0130	0.0110
LogisticRegression	0.0589	0.1042	0.1234	0.7938	0.8240	0.8344
SVC	0.1628	0.1898	0.1604	1.5649	1.3771	1.3968
LinearSVC	0.0577	0.0884	0.1109	0.5096	0.6259	0.6231
NearestCentroid	0.1596	0.2173	0.1927	0.0090	0.0110	0.0100
CalibratedClassifierCV	0.0524	0.0809	0.1010	2.5534	2.7130	2.7518
SGDClassifier	0.0623	0.0980	0.1540	0.1057	0.1207	0.1099
Perceptron	0.0211	0.1231	0.1019	0.0529	0.0595	0.0562
PassiveAggressiveClassifier	0.0348	0.0973	0.0827	0.0549	0.0598	0.0611
BernoulliNB	0.0386	0.0732	0.0754	0.0110	0.0110	0.0120
RidgeClassifier	0.0321	0.0478	0.0757	0.0180	0.0199	0.0200
RidgeClassifierCV	0.0321	0.0478	0.0757	0.0379	0.0239	0.0209
AdaBoostClassifier	0.0389	0.1004	0.0936	0.4220	0.4199	0.4144
DummyClassifier	0.0256	0.0256	0.0256	0.0070	0.0080	0.0080
QuadraticDiscriminantAnalysis	0.1194	0.2193	0.1941	0.0150	0.0130	0.0120

Table 5.21 Performance of evaluation metric (balanced accuracy) for imputation data using UMAP with 3 components and different near-neighbor values using “train_test_split”, along with the time taken for training each algorithm.

Classification Algorithm	Balanced Accuracy		Time Taken (seconds)		
	UMAP10-3	UMAP30-3	UMAP123-3	UMAP10-3	UMAP30-3 UMAP123-3
ExtraTreesClassifier	0.8501	0.8152	0.7342	0.3680	0.4879 0.5206
LGBMClassifier	0.8059	0.0527	0.6550	1.5523	0.8830 2.0156
RandomForestClassifier	0.8327	0.7849	0.6810	0.6910	0.7508 0.9259
DecisionTreeClassifier	0.8270	0.7603	0.6591	0.0250	0.0359 0.0329
BaggingClassifier	0.8035	0.7696	0.6690	0.1192	0.1676 0.1750
ExtraTreeClassifier	0.8102	0.7291	0.6009	0.0100	0.0130 0.0130
LabelPropagation	0.4042	0.2543	0.2140	0.6273	0.6401 0.6611
LabelSpreading	0.4047	0.2541	0.1995	1.0045	1.0271 1.0150
KNeighborsClassifier	0.6502	0.6539	0.5882	0.0379	0.0399 0.0399
GaussianNB	0.1618	0.1966	0.2085	0.0209	0.0279 0.0289
LinearDiscriminantAnalysis	0.0987	0.1297	0.1382	0.0100	0.0100 0.0139
LogisticRegression	0.1051	0.1389	0.1403	0.7938	0.7934 1.0130
SVC	0.2495	0.2222	0.1655	1.5649	1.3185 1.3388
LinearSVC	0.0696	0.1279	0.1383	0.5096	1.0410 0.8294
NearestCentroid	0.1851	0.2014	0.2403	0.0090	0.0100 0.0169
CalibratedClassifierCV	0.0696	0.1152	0.1161	2.5534	4.5479 3.8081
SGDClassifier	0.0678	0.1025	0.1533	0.1057	0.1563 0.1351
Perceptron	0.0787	0.1017	0.1128	0.0529	0.0792 0.0957
PassiveAggressiveClassifier	0.0532	0.1223	0.1028	0.0549	0.0607 0.0947
BernoulliNB	0.0459	0.0761	0.1018	0.0110	0.0120 0.0120
RidgeClassifier	0.0647	0.0758	0.1018	0.0180	0.0189 0.0199
RidgeClassifierCV	0.0637	0.0758	0.1018	0.0379	0.0209 0.0200
AdaBoostClassifier	0.1124	0.0762	0.0761	0.4220	0.4830 0.4514
DummyClassifier	0.0256	0.0256	0.0256	0.0070	0.0100 0.0070
QuadraticDiscriminantAnalysis	0.2144	0.2767	0.2725	0.0150	0.0160 0.0200

Table 5.22 Performance of evaluation metric (balanced accuracy) for imputation data using the PCA method for different components using k-fold cross validation, along with time taken for training each algorithm.

Classification Algorithm	Balanced Accuracy			Time Taken (seconds)		
	PCA-2	PCA-3	PCA-4	PCA-2	PCA-3	PCA-4
ExtraTreesClassifier	0.7769	0.8358	0.8709	0.4277	0.4301	0.4136
RandomForestClassifier	0.7451	0.8253	0.8577	0.7073	0.7143	1.0609
BaggingClassifier	0.7012	0.7793	0.8319	0.1218	0.1573	0.1886
DecisionTreeClassifier	0.7308	0.7820	0.8267	0.0228	0.0286	0.0344
ExtraTreeClassifier	0.7311	0.7674	0.8240	0.0104	0.0097	0.0100
LabelPropagation	0.1994	0.3401	0.5514	0.6835	0.7046	0.7152
LabelSpreading	0.1932	0.3267	0.5311	1.3347	1.3334	1.3164
KNeighborsClassifier	0.4742	0.5219	0.5683	0.0334	0.0362	0.0377
LGBMClassifier	0.7255	0.8084	0.5301	1.7637	1.8761	1.5696
LinearDiscriminantAnalysis	0.1332	0.1479	0.2008	0.0098	0.0106	0.0113
GaussianNB	0.1361	0.2129	0.3126	0.0104	0.0110	0.0112
LogisticRegression	0.1109	0.1298	0.2023	0.7755	0.7843	0.7778
SVC	0.1442	0.1816	0.2498	1.1508	1.1726	1.1509
LinearSVC	0.0610	0.0670	0.0917	1.2491	2.0262	2.1846
NearestCentroid	0.1630	0.1893	0.2534	0.0084	0.0084	0.0086
CalibratedClassifierCV	0.0561	0.0661	0.0812	5.1969	8.2681	8.6005
SGDClassifier	0.0741	0.0789	0.1144	0.1237	0.1397	0.1587
QuadraticDiscriminantAnalysis	0.1475	0.2270	0.4012	0.0106	0.0116	0.0116
Perceptron	0.0725	0.0650	0.0869	0.0577	0.0619	0.0651
PassiveAggressiveClassifier	0.0595	0.0611	0.0777	0.0596	0.0639	0.0696
BernoulliNB	0.0503	0.0503	0.0503	0.0093	0.0090	0.0093
RidgeClassifier	0.0496	0.0490	0.0481	0.0115	0.0114	0.0122
RidgeClassifierCV	0.0496	0.0490	0.0481	0.0203	0.0211	0.0212
AdaBoostClassifier	0.1009	0.1008	0.0865	0.4045	0.4412	0.4750
DummyClassifier	0.0256	0.0256	0.0256	0.0058	0.0061	0.0060

Table 5.23 Performance of evaluation metrics (balanced accuracy) for imputation data using t-SNE with 2 components and different perplexity values using k-fold cross validation, along with the time taken for training each algorithm.

Classification Algorithm	Balanced Accuracy		Time Taken (seconds)			
	Tsne10-2	Tsne30-2	Tsne123-2	Tsne10-2	Tsne30-2	Tsne123-2
Extra TreesClassifier	0.8921	0.8929	0.8722	0.3265	0.3241	0.3290
RandomForestClassifier	0.8886	0.8835	0.8628	0.6431	0.6288	0.6483
BaggingClassifier	0.8669	0.8621	0.8373	0.1090	0.1071	0.1132
Decision TreeClassifier	0.8863	0.8815	0.8514	0.0208	0.0204	0.0219
Extra TreeClassifier	0.8753	0.8700	0.8470	0.0088	0.0086	0.0090
LabelPropagation	0.3385	0.2772	0.2379	0.6733	0.6728	0.6687
LabelSpreading	0.3294	0.2734	0.2304	1.3099	1.2979	1.2775
KNeighborsClassifier	0.6750	0.6863	0.6650	0.0328	0.0333	0.0332
LGBMClassifier	0.5336	0.7035	0.8494	1.5011	1.6371	1.7765
LinearDiscriminantAnalysis	0.0656	0.1204	0.1369	0.0099	0.0098	0.0096
GaussianNB	0.1334	0.1789	0.1535	0.0104	0.0105	0.0100
LogisticRegression	0.0740	0.1223	0.1377	0.7673	0.7828	0.7868
SVC	0.2208	0.2254	0.1828	1.1766	1.0702	1.0515
LinearSVC	0.0643	0.0488	0.1015	0.4130	0.3105	0.3483
NearestCentroid	0.1886	0.2232	0.1755	0.0082	0.0087	0.0083
CalibratedClassifierCV	0.0517	0.0574	0.0903	1.9721	1.5766	1.6991
SGDClassifier	0.0739	0.0949	0.1271	0.1121	0.1167	0.1138
QuadraticDiscriminantAnalysis	0.1271	0.2039	0.2138	0.0103	0.0108	0.0106
Perceptron	0.0630	0.0751	0.0775	0.0552	0.0553	0.0556
PassiveAggressiveClassifier	0.0777	0.0642	0.0660	0.0577	0.0590	0.0584
BernoulliNB	0.0566	0.0445	0.0710	0.0089	0.0092	0.0089
RidgeClassifier	0.0393	0.0417	0.0505	0.0119	0.0116	0.0116
RidgeClassifierCV	0.0393	0.0416	0.0505	0.0208	0.0208	0.0202
AdaBoostClassifier	0.0680	0.0643	0.0739	0.3950	0.3911	0.3930
DummyClassifier	0.0256	0.0256	0.0258	0.0060	0.0061	0.0056

Table 5.24 Performance of evaluation metric (balanced accuracy) for imputation data using t-SNE with 3 components and different perplexity values using k-fold cross validation, along with the time taken for training each algorithm.

Classification Algorithm	Balanced Accuracy			Time Taken (seconds)		
	Tsne10-3	Tsne30-3	Tsne123-3	Tsne10-3	Tsne30-3	Tsne123-3
ExtraTreesClassifier	0.9020	0.8949	0.8782	0.3401	0.3438	0.3352
RandomForestClassifier	0.8931	0.8874	0.8698	0.6713	0.6871	0.6670
BaggingClassifier	0.8700	0.8680	0.8491	0.1562	0.1625	0.1540
DecisionTreeClassifier	0.8859	0.8789	0.8594	0.0282	0.0291	0.0284
ExtraTreeClassifier	0.8596	0.8642	0.8472	0.0096	0.0092	0.0090
LabelPropagation	0.6281	0.5808	0.5165	0.6709	0.6964	0.6811
LabelSpreading	0.6157	0.5648	0.4993	1.2951	1.3629	1.3262
KNeighborsClassifier	0.7082	0.6895	0.6872	0.0350	0.0353	0.0349
LGBMClassifier	0.5525	0.8778	0.2000	1.5646	1.8609	1.2253
LinearDiscriminantAnalysis	0.1147	0.1780	0.1422	0.0104	0.0109	0.0103
GaussianNB	0.1871	0.2149	0.1996	0.0106	0.0112	0.0109
LogisticRegression	0.1378	0.1809	0.1572	0.8007	0.8137	0.7964
SVC	0.3073	0.2797	0.2390	1.1099	1.0340	1.0214
LinearSVC	0.0845	0.1382	0.1246	0.4434	0.3702	0.4347
NearestCentroid	0.2421	0.2732	0.2344	0.0086	0.0088	0.0083
CalibratedClassifierCV	0.0847	0.1343	0.1344	2.0596	1.7766	1.9859
SGDClassifier	0.0899	0.1176	0.1127	0.1249	0.1255	0.1270
QuadraticDiscriminantAnalysis	0.2680	0.2825	0.2649	0.0116	0.0116	0.0112
Perceptron	0.0893	0.1057	0.0892	0.0555	0.0563	0.0571
PassiveAggressiveClassifier	0.0835	0.1015	0.0978	0.0623	0.0652	0.0604
BernoulliNB	0.0451	0.0631	0.0794	0.0092	0.0091	0.0090
RidgeClassifier	0.0438	0.0597	0.0514	0.0114	0.0124	0.0120
RidgeClassifierCV	0.0437	0.0597	0.0514	0.0209	0.0217	0.0209
AdaBoostClassifier	0.0680	0.0532	0.0632	0.4350	0.4517	0.4383
DummyClassifier	0.0256	0.0256	0.0256	0.0064	0.0062	0.0060

Table 5.25 Performance of evaluation metrics (balanced accuracy) for imputation data using UMAP with 2 components and different near-neighbor values using k-fold cross validation, along with the time taken for training each algorithm.

Classification Algorithm	Balanced Accuracy		Time Taken (seconds)		
	UMAP10-2	UMAP30-2	UMAP123-2	UMAP10-2	UMAP30-2 UMAP123-2
ExtraTreesClassifier	0.8110	0.7424	0.6083	0.4148	0.4404 0.5185
RandomForestClassifier	0.7918	0.6940	0.5798	0.6966	0.6987 0.7151
BaggingClassifier	0.7718	0.6826	0.5693	0.1154	0.1199 0.1241
DecisionTreeClassifier	0.7774	0.6911	0.5701	0.0220	0.0232 0.0238
ExtraTreeClassifier	0.7567	0.6468	0.5354	0.0098	0.0098 0.0102
LabelPropagation	0.3291	0.2181	0.1495	0.6906	0.6815 0.6858
LabelSpreading	0.3192	0.2117	0.1475	1.3000	1.3359 1.3080
KNeighborsClassifier	0.6586	0.6123	0.5288	0.0332	0.0334 0.0340
LGBMClassifier	0.7266	0.3032	0.2506	1.8212	1.3384 1.3348
LinearDiscriminantAnalysis	0.0597	0.1092	0.1059	0.0098	0.0097 0.0098
GaussianNB	0.0837	0.1664	0.1777	0.0104	0.0104 0.0106
LogisticRegression	0.0597	0.1090	0.1301	0.7845	0.7887 0.7860
SVC	0.1746	0.1861	0.1589	1.3408	1.0326 1.0518
LinearSVC	0.0572	0.0889	0.1008	0.6045	0.4664 0.4674
NearestCentroid	0.1542	0.2195	0.1912	0.0084	0.0084 0.0086
CalibratedClassifierCV	0.0515	0.0826	0.0980	2.7329	2.1119 2.1660
SGDClassifier	0.0623	0.0999	0.1255	0.1183	0.1174 0.1116
QuadraticDiscriminantAnalysis	0.1039	0.2194	0.1824	0.0106	0.0106 0.0109
Perceptron	0.0439	0.0973	0.0933	0.0555	0.0573 0.0559
PassiveAggressiveClassifier	0.0478	0.1005	0.1054	0.0589	0.0584 0.0593
BernoulliNB	0.0384	0.0729	0.0756	0.0094	0.0088 0.0090
RidgeClassifier	0.0323	0.0478	0.0755	0.0114	0.0116 0.0116
RidgeClassifierCV	0.0323	0.0478	0.0755	0.0202	0.0202 0.0208
AdaBoostClassifier	0.0750	0.1011	0.0828	0.4024	0.4058 0.4120
DummyClassifier	0.0256	0.0256	0.0256	0.0060	0.0062 0.0058

Table 5.26 Performance of evaluation metrics (balanced accuracy) for imputation data using UMAP with 3 components and different near-neighbor values using k-fold cross validation, along with the time taken for training each algorithm.

Classification Algorithm	Balanced Accuracy			Time Taken (seconds)		
	UMAP10-3	UMAP30-3	UMAP123-3	UMAP10-3	UMAP30-3	UMAP123-3
ExtraTreesClassifier	0.8364	0.7895	0.7095	0.4025	0.4238	0.4366
RandomForestClassifier	0.8125	0.7496	0.6620	0.7047	0.6986	0.7160
BaggingClassifier	0.7875	0.7276	0.6565	0.1689	0.1582	0.1657
DecisionTreeClassifier	0.7928	0.7254	0.6328	0.0317	0.0290	0.0304
ExtraTreeClassifier	0.7730	0.6983	0.6043	0.0094	0.0098	0.0098
LabelPropagation	0.4010	0.2603	0.2189	0.6862	0.6830	0.6763
LabelSpreading	0.3954	0.2556	0.2128	1.3110	1.3243	1.3079
KNeighborsClassifier	0.6526	0.6310	0.6115	0.0346	0.0342	0.0344
LGBMClassifier	0.7635	0.4674	0.6154	1.8766	1.5890	1.8907
LinearDiscriminantAnalysis	0.1032	0.1261	0.1419	0.0103	0.0102	0.0102
GaussianNB	0.1583	0.2012	0.1972	0.0108	0.0108	0.0106
LogisticRegression	0.1041	0.1424	0.1455	0.7614	0.7653	0.7951
SVC	0.2490	0.2245	0.1585	1.1175	0.9994	1.0043
LinearSVC	0.0725	0.1315	0.1385	0.7941	0.8242	0.7081
NearestCentroid	0.2129	0.1877	0.2450	0.0082	0.0086	0.0086
CalibratedClassifierCV	0.0698	0.1198	0.1306	3.3967	3.4009	3.0388
SGDClassifier	0.0881	0.1174	0.1538	0.1399	0.1354	0.1227
QuadraticDiscriminantAnalysis	0.2310	0.2862	0.2889	0.0115	0.0116	0.0112
Perceptron	0.0788	0.1166	0.0929	0.0595	0.0622	0.0590
PassiveAggressiveClassifier	0.0832	0.1077	0.0796	0.0622	0.0648	0.0624
BernoulliNB	0.0459	0.0759	0.1015	0.0090	0.0094	0.0092
RidgeClassifier	0.0619	0.0739	0.1017	0.0116	0.0120	0.0119
RidgeClassifierCV	0.0615	0.0738	0.1017	0.0204	0.0218	0.0205
AdaBoostClassifier	0.0714	0.0755	0.0761	0.4289	0.4329	0.4294
DummyClassifier	0.0256	0.0256	0.0256	0.0059	0.0060	0.0062

clustering techniques applied to fourteen operational and design variables. These indices distill complex multidimensional data into interpretable metrics that classify vessels into distinct performance categories based on environmental impact, physical characteristics, and operational patterns. The resulting graphical representation provides stakeholders with transparent, actionable information about vessel environmental performance, distinguishing between ships of varying sizes and emission profiles. Second, the research demonstrates that M classification can accurately identify ship engine types from operational and physical data with over ninety-five percent accuracy, enabling strategic optimization of engine selection during vessel design and throughout operational lifecycles.

The practical implications of these methodologies extend across multiple dimensions of maritime operations. For shipowners and operators, accurate engine classification facilitates informed decisions regarding fleet composition, maintenance strategies, and compliance with environmental regulations. By identifying optimal engine models based on vessel characteristics and intended operational profiles, the framework supports both cost reduction and emission minimization. The ability to validate manufacturer specifications against observed operational performance enhances regulatory oversight and ensures accurate emissions inventories. Furthermore, these approaches enable the development of digital twin applications and RTM systems that can evolve from static assessments toward dynamic decision-support tools capable of detecting operational anomalies and optimizing performance continuously.

From a broader industry perspective, the integration of environmental performance indices with engine classification capabilities creates a comprehensive framework for advancing maritime sustainability. The transparency afforded by visual performance tags combined with validated engine selection methodologies supports both competitive differentiation and collective progress toward environmental objectives. As data collection infrastructure expands to include higher temporal resolution measurements, these frameworks will enable identification of seasonal patterns, early detection of performance degradation, and adaptive management strategies that respond to changing operational conditions and regulatory requirements.

The primary limitation of this research stems from incomplete coverage of all available engine models and missing values in operational efficiency variables within the available dataset. While imputation techniques produced reliable estimates, more comprehensive data collection would enhance model robustness and generalizability. Nevertheless, the methodologies presented successfully identify underlying patterns within current industry practices, validating effective strategies while providing ac-

cessible pathways for broader application of best practices during vessel design and operation.

In conclusion, this research demonstrates that M techniques can simultaneously advance environmental assessment and operational optimization in maritime contexts, providing integrated solutions that reconcile economic efficiency with environmental responsibility. By transforming complex operational data into interpretable indices and actionable classifications, these methodologies establish a foundation for evidence-based decision-making that supports regulatory compliance, operational excellence, and the transition toward sustainable shipping practices. As the industry continues to evolve under intensifying environmental pressures, such data-driven frameworks will prove essential for achieving the dual objectives of economic competitiveness and ecological stewardship.

5.6 Summary

This chapter develops and evaluates M methodologies for dynamic maritime environmental monitoring, focusing on two interconnected applications: assessing vessel environmental performance and classifying ship engine models. The research addresses the critical need for data-driven tools that support regulatory compliance, operational efficiency, and sustainability in the maritime industry.

The environmental performance assessment framework integrates real-time data streams from onboard sensors capturing fuel consumption, CO emissions, engine status, and navigational parameters. By applying PCA to 14 operational and design variables from 2198 General Cargo ships, three key indices emerge: (1) Ship Physical Dimensions, (2) Operational Environmental Efficiency, and (3) Operating Region. K-means clustering identifies four distinct vessel categories: large environmentally friendly ships, small environmentally friendly ships, large non-environmentally friendly ships, and small non-environmentally friendly ships. A novel graphical tagging system visualizes these performance indices, providing intuitive at-a-glance environmental impact assessment. Validation using 62 Container ships confirms the robustness and transferability of the derived indices across vessel types.

The engine model classification component addresses a critical data gap in maritime datasets where engine specifications are often missing or unreliable due to privacy concerns and limited data availability. Using a comprehensive dataset of 7159 observations across 39 engine models, the study implements a four-step supervised learning pipeline: preprocessing, MICE for missing data, dimensionality reduction (PCA, UMAP, t-SNE),

and classification using 25 different algorithms. The `ExtraTreesClassifier` achieves the highest performance with 95.07% balanced accuracy on imputed data, demonstrating that operational and physical characteristics provide sufficient information to reliably identify engine models. Comparative analysis reveals that DT-based classifiers (`ExtraTreesClassifier`, `RandomForestClassifier`, `BaggingClassifier`) consistently outperform linear models and other approaches, with t-SNE emerging as the most effective dimensionality reduction method.

The research addresses significant data quality challenges inherent in maritime operations. The MICE imputation procedure, validated through rigorous statistical testing (Little's test, F-tests, MSE, RMSE, MAE metrics), successfully handles missing values ranging from 20% to 70% across variables. Three gradient boosting regressors (Light Gradient Boosting, Extreme Gradient Boosting, Bayesian Ridge) are compared, with Extreme Gradient Boosting yielding the most accurate imputations.

Practical implications include support for optimal vessel design, engine selection strategies, digital twin development, performance monitoring, and predictive maintenance. The high classification accuracy enables validation of manufacturer-reported specifications, detection of potential misreporting, and verification of fuel consumption and emissions characteristics under real operating conditions. The continuous, adaptive nature of the monitoring framework supports strategic fleet management decisions, promoting safer, greener, and more cost-effective maritime transportation while meeting tightening global environmental regulations.

Chapter 6

Monitoring Applications in Hydrological Data: LSTM-Based Precipitation Forecasting

6.1 Introduction

The imperative to mitigate global climate change, driven primarily by the burning of fossil fuels for electricity and heat production, has accelerated the adoption of renewable energy sources. This transition is legally mandated within the European Union by the Renewable Energy Directive (2009/28/EC), which was revised in 2018 and has been legally binding since June 2021. Hydropower serves as the backbone of this effort, providing almost half of the world's low-carbon electricity today and positioned to contribute significantly to the EU's 2030 energy targets. Unlike more intermittent renewable sources such as solar and wind, hydropower offers crucial flexibility and storage capacity, which are vital for maintaining the stability of the modern electricity system.

The design and operational viability of any hydropower station are fundamentally determined by regional topographic and meteorologic characteristics, with available water resources, expressed as runoff, being the critical factor. Runoff is typically estimated using historical precipitation timeseries. Under the growing effects of climate change—which include altered water availability and an increase in extreme weather events—accurate precipitation prediction has become essential. Such forecasts are crucial not only for reassessing the capacity and resilience of existing hydropower stations but also for reliably evaluating the potential of new plants.

To this end, modern environmental monitoring increasingly relies on integrating continuous, real-time data streams with advanced analytical techniques. However, data from these monitoring systems are often incomplete, noisy, and temporally dependent. Consequently, rigorous preprocessing steps, such as imputation and dimensionality reduction, are prerequisites for transforming this raw data into actionable insights.

While traditional time series models like ARIMA and SARIMA have been used extensively for precipitation forecasting, ML techniques offer a compelling and increasingly popular alternative for hydrometeorological prediction. Among these, ANNs have demonstrated considerable success in producing accurate rainfall predictions in hydrology. Various ANN architectures have been successfully applied to forecast precipitation probability and quantity, predict other meteorological events, and estimate power generation from photovoltaic and wind turbine systems. A thorough literature review of ML models for rainfall prediction is provided in [284]. Recurrent neural networks like Seasonal Long Short-Term Memory (SLSTM) are particularly well-suited for this domain, as they excel at capturing the long-term dependencies, seasonality, and nonlinearities inherent in temporal data streams.

In the present chapter, a complete procedure is proposed to obtain accurate predictions of meteorological data, specifically precipitation, to support hydropower production. This methodology is applied to data from the Greek automated weather stations network in the Peloponnese and the Ionian islands—a region highly dependent on renewable energy.

The proposed prediction algorithm successfully identified climatic zones based on geographic and climatic characteristics for most stations, leading to realistic precipitation predictions, though it did underestimate total annual precipitation for some stations, a weakness also reported in other research. A diagram of the complete procedure for obtaining accurate meteorological predictions to support hydropower production. Continuous data streams from the Greek automated weather stations network are first processed through a comprehensive monitoring scenario that includes data quality evaluation, filtering, and both spatial and temporal imputation. The resulting complete dataset is then used to train a SLSTM model, which generates precipitation forecasts (see Figure 6.1) used for adaptive hydropower grid management and planning.

In line with recent evidence, the fusion of dense station telemetry with DL substantially improves precipitation nowcasting and short-term hydrometeorological forecasts, thereby strengthening operational planning for hydro assets [18]. Following this paradigm, the present chapter preprocesses continuous station streams and trains a seasonal LSTM to yield forecasts aligned with grid-operation horizons.

The synergy between real-time data acquisition and adaptive ML models, as described herein, enables the creation of a dynamic feedback loop for hydropower management. By continuously processing a live stream of meteorological data, the system can generate up-to-the-minute precipitation forecasts. This empowers grid operators to make responsive adjustments to the energy mix, transforming static predictions into a continuous, adaptive planning tool and ensuring grid stability.

This chapter is structured as follows: Section 6.2 presents the study area and highlights the need for accurate precipitation predictions. Section 6.3 describes the methodology used for the analysis. In Section 6.4, the results of the analysis of the available data is presented, while Section 6.5 provides concluding remarks.

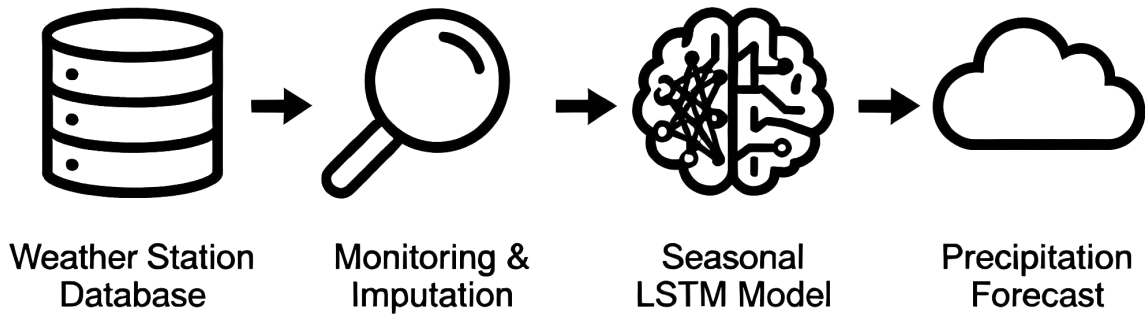


Fig. 6.1 A conceptual diagram of the ML framework for predicting annual rainfall to support hydropower management in Greece.

6.2 Motivation

The European Union's energy policy is driven by the recast Renewable Energy Directive 2018/2001/EU, which mandates an accelerated integration of renewable sources to achieve a 55% reduction in net greenhouse gas emissions by 2030 and climate neutrality by 2050. This objective has been further amplified by measures designed to rapidly diminish the EU's dependence on Russian fossil fuels. Within this context, the EU Commission has identified an imperative role for hydropower in Europe's energy transition. This is corroborated by the International Energy Agency (IEA), which, in its first Hydropower Special Market Report, predicts that Europe's total installed hydropower capacity will increase by approximately 8% by 2030. To safeguard and increase the reliability of this contribution, especially under the effects of climate change, the accurate prediction of monthly precipitation is essential.

This research focuses on the Peloponnese and the Ionian islands in Greece, a region of significant strategic importance for several reasons. First, the area is a major socioeconomic hub, home to circa 1.4 million permanent residents (13% of Greece’s population) and attracting over 16% of the country’s tourism (circa 2.6 million visits in 2021), according to InSETE (<https://insete.gr/districts/?lang=en#>, accessed on 20 December 2022). Second, from an energy perspective, the regions of Peloponnese, Western Greece, and the Ionian islands have an installed renewable capacity of circa 2 GW, accounting for over 21% of Greece’s total of 9.2 GW. This makes the region critical to the stability of the national electricity system. Third, the expansion of the national transmission system to an ultra-high voltage 400 kV network, the future backbone of the Hellenic energy system, positions the Peloponnese as a key energy node with planned interconnections to Crete, Attica, and Northern Greece.

At the national level, Greece’s total installed hydropower capacity is 3217.4 MW, with an average annual production of approximately 4020 GWh, which covers 8% to 10% of the country’s total energy demand, depending on the hydrological year. The necessity of accurate precipitation forecasting is therefore evident for maintaining stable energy production. Within the specific region examined in this study, there is one large hydropower station (Ladon, 70 MW) and numerous small hydropower stations with a combined capacity of circa 58 MW, directly linking the regional meteorological conditions to a tangible portion of the national energy supply.

6.3 Materials and Methods

Let the dataset consist of monthly meteorological records, such as average surface temperature or total precipitation, from m automated weather stations. An observation from station i at time t is denoted by x_{it} , where $i = 1, 2, \dots, m$ and $t = 1, 2, \dots, n_i$. Each x_{it} typically represents an aggregation (e.g., average or sum) of daily measurements.

Real-world meteorological datasets of this nature are characteristically heterogeneous and incomplete. The time series often exhibit strong seasonal patterns and contain missing values. Furthermore, since the operational periods and data completeness vary significantly across stations, the resulting time series, x_i , are of unequal length, n_i . These irregularities preclude the direct application of most standard ML algorithms for forecasting.

Consequently, a rigorous and carefully designed preprocessing pipeline is a critical prerequisite. Such a pipeline is necessary to systematically handle missing data and harmonize the dataset for subsequent analysis. The following subsections detail the

proposed preprocessing steps and then introduce the ML algorithms employed for prediction.

6.3.1 Data Preprocessing

Meteorological datasets are frequently characterized by heterogeneity in temporal coverage and significant data sparsity. These issues stem from several operational realities. For instance, national meteorological services may deploy stations for temporary testing or expand their networks incrementally, leading to varied operational timelines across the network. Within a given station's record, missing values often arise from staggered commissioning dates, equipment failures (particularly during extreme weather events), or measurement errors.

Addressing these challenges requires a systematic approach. As a preliminary data-cleaning step, stations with insufficient data are excluded from the analysis. The criteria for exclusion include a short operational duration, inactivity by the end of the study period, or an exceptionally high proportion of missing values, which may indicate intermittent or testing-phase usage.

For the remaining stations, a sophisticated imputation strategy is necessary. This strategy is informed by an understanding of the data's underlying structure, which can be elucidated from key ancillary information. This includes each station's geographical metadata (longitude, latitude, and altitude), its proximity to neighboring stations, its designated climate zone, and the inherent temporal structures (e.g., seasonality and long-term memory) within the data itself. Leveraging these characteristics, a two-step imputation procedure is proposed, which is detailed in the subsequent section.

Spatial Imputation

The first step in our preprocessing pipeline addresses missing data through a spatially-aware imputation method. Although monthly or annual aggregates of meteorological data often exhibit strong spatial correlation among nearby stations, this assumption can be violated in regions with complex geography, such as the rugged relief and varied coastline of Greece. In such areas, local factors, particularly elevation, can cause significant variation in meteorological measurements like total precipitation, even between geographically proximate stations.

To account for this localized variability, our methodology incorporates the concept of **climate zones**. The core principle is to impute missing values by borrowing data only from neighboring stations that belong to the same climate zone. This ensures that the

imputed values are drawn from a comparable climatological context. These zones can be obtained either from established classifications provided by national meteorological services or derived empirically by applying a suitable clustering algorithm to the available station data.

To operationalize the concept of a “nearest station” within a given climate zone, we employ the **Haversine distance**. This metric calculates the great-circle distance between two points on a sphere, making it well-suited for determining the shortest distance between meteorological stations. The distance d is given by :

$$d = R \cdot \theta$$

where R is the Earth’s radius ($R = 6378137$ m) and θ is the central angle between two stations, A and B, with coordinates (latitude, longitude) (ϕ_A, ℓ_A) and (ϕ_B, ℓ_B) . The angle θ is determined as follows:

$$\cos \theta = \sin \phi_A \cdot \sin \phi_B + \cos \phi_A \cdot \cos \phi_B \cdot \cos \Delta \ell$$

where $\Delta \ell = \ell_A - \ell_B$. The imputation procedure for a missing value x_{it} at station i is as follows:

1. All other stations j within the same cluster (where there are n_c stations) are ranked based on their Haversine distance d_{ij} from station i .
2. The algorithm iterates through this ranked list, starting with the nearest station $((j), \text{ for } j = 1, \dots, n_c - 1)$.
3. The missing value x_{it} is replaced with the corresponding value $x_{(j)t}$ from the first station in the sequence that has a non-missing observation for that time point.
4. This search is constrained by a predefined distance threshold; if no suitable donor station is found within this radius, the value remains missing to avoid borrowing information from geographically dissimilar locations.

Temporal Imputation

The second step of the imputation procedure addresses any missing values that persist after the spatial imputation phase. Such gaps typically remain when, for a given time point, no suitable data could be borrowed from a neighboring station within the predefined distance threshold. To fill these remaining missing values, we employ a

temporal imputation method that leverages the inherent structure of each station's individual time series.

Specifically, we use an imputation approach based on a **structural time series model and Kalman smoothing**. Before applying the model, a data transformation, such as a log transformation, can be applied. This is a common and often necessary practice for meteorological variables like precipitation, as it serves to stabilize the variance and reduce the positive skewness frequently observed in the data's distribution.

The imputation process itself involves two primary stages:

1. A structural model, designed to capture underlying components like trend and seasonality, is fitted to the available observations of each time series using maximum likelihood estimation.
2. The Kalman filter is then applied to this fitted model to generate optimal estimates for the missing values, conditioning on all observed data within the series.

This approach is well-established for handling missing data in time series contexts. A similar methodology has been applied in [285]. For a comprehensive technical treatment of the method, the reader is referred to [286].

Potential Enhancement through Regression-Adjusted Monitoring for Station Networks

The spatial correlation structure among neighboring meteorological stations within climate zones, as evidenced by the successful spatial imputation strategy, suggests potential benefits from regression-adjusted monitoring approaches for real-time anomaly detection. While not implemented in the current forecasting framework, regression-adjusted control charts [59] could complement the spatial imputation by providing early warning of sensor malfunctions or anomalous local weather events. By modeling expected precipitation patterns at each station as a function of its neighbors' measurements, deviations from these regression relationships could signal data quality issues or localized meteorological phenomena requiring immediate attention. Future integration of such monitoring capabilities could enhance the reliability of real-time data streams feeding into the SLSTM prediction pipeline, particularly for detecting systematic biases or sensor drift that may compromise forecast accuracy over extended operational periods.

6.3.2 Machine Learning Prediction Algorithm

Artificial Neural Networks (ANNs) have emerged as powerful models for a wide array of tasks, including classification, clustering, pattern recognition, and prediction across numerous fields. Their application to weather forecasting has grown significantly in recent decades. Initial research often focused on establishing linear relationships between input weather data and target variables. However, with the recognition of inherent nonlinearity in meteorological systems, the focus has shifted toward nonlinear prediction methods. While various nonlinear statistical models exist, most require the model's functional form to be specified a priori. Given that meteorological data exhibit both nonlinearity and highly irregular trends, ANNs provide a superior, data-driven alternative for identifying the complex structural relationships between different components.

A particularly advanced and suitable architecture for this task is the LSTM network, a type of recurrent neural network (RNN) specifically designed to learn long-term dependencies in sequence prediction problems. Traditional RNNs often struggle to retain the influence of early observations as the input sequence length increases, a problem known as the vanishing gradient problem. The specialized structure of LSTMs effectively counterbalances this disadvantage.

LSTMs are composed of a chain of repeating modules, or cells, which act as memory blocks. The flow of information into, within, and out of these cells is regulated by a system of three gates:

- The **forget gate** determines which information from the previous cell state should be discarded.
- The **input gate** controls which new information is stored in the current cell state.
- The **output gate** determines what information from the cell state is used to generate the output for the current time step.

The flow of information through these gates is managed by activation functions, most commonly the sigmoid function. The sigmoid function outputs values between 0 and 1, acting as a switch: values close to 1 permit a large proportion of information to pass through the gate, whereas values close to 0 restrict the flow. In a typical LSTM network, the sigmoid and hyperbolic tangent (*tanh*) activation functions are used in combination to update and pass information. The specific formulas for the forget gate

(f_t), input gate (i_t), output gate (o_t), candidate cell state (c'_t), cell state (c_t), and hidden state (h_t) are as follows:

$$\begin{aligned}
 f_t &= \sigma(w_{xf}x_t + w_{hf}h_{t-1} + b_f) \\
 i_t &= \sigma(w_{xi}x_t + w_{hi}h_{t-1} + b_i) \\
 c'_t &= \tanh(w_{xc'}x_t + w_{hc'}h_{t-1} + b_{c'}) \\
 o_t &= \sigma(w_{xo}x_t + w_{ho}h_{t-1} + b_o) \\
 c_t &= \sigma(f_t c_{t-1} + i_t c'_t) \\
 h_t &= o_t * \tanh(c_t)
 \end{aligned}$$

In the previously mentioned gate equations, the terms w_{xf} , w_{xi} , $w_{xc'}$, and w_{xo} represent the weight matrices applied to the input x_t , while w_{hf} , w_{hi} , and $w_{hc'}$ are the weight matrices applied to the previous hidden state h_{t-1} . The operator $*$ denotes element-wise multiplication. Additionally, b_f , b_i , $b_{c'}$, and b_o are the bias vectors for each respective gate.

The ability of LSTMs to capture long-term dependencies is a critical feature for time series analysis. Time series data often exhibit not only short-term memory (dependence on recent observations) but also persistent, long-range behaviors arising from seasonality and trend. While the trend component can often be handled by conditioning on recent observations, capturing seasonality can be challenging if the input sequence provided to a standard LSTM network is not sufficiently long to span multiple seasonal cycles.

To overcome this limitation, an important extension is the SLSTM network, as proposed by [287]. This architecture explicitly incorporates seasonality attributes into the model structure.

A general representation of the feature vector $\mathbf{X}$ and target vector $\mathbf{Y}$ for an SLSTM network is shown in Table 6.1. For a time series with n available observations, the feature vector is constructed assuming a seasonality period of s , using k seasonal lags and the $m+1$ most recent values. The corresponding target vector is constructed from the subsequent f observations. Figure 6.2 provides a simple graphical representation of the first feature and target vectors for a specific case where $s = 12$, $k = 2$, $m = 11$, and $f = 12$.

Table 6.1 General representation of feature vector $\mathbf{X}$ and target values $\mathbf{Y}$ for an SLSTM network. Here, n is the total number of observations, s is the seasonal period, k the number of seasonal lags, m the max lagged values, and f the forecast horizon.

No.	$\mathbf{X}$	$\mathbf{Y}$
1	$[x_1, x_{1+s}, \dots, x_{1+ks}, \dots, x_{1-m}, \dots, x_1]$	$[x_2, \dots, x_{1+f}]$
2	$[x_2, x_{2+s}, \dots, x_{2+ks}, \dots, x_{2-m}, \dots, x_2]$	$[x_3, \dots, x_{2+f}]$
$\vdots$	$\vdots$	$\vdots$
t	$[x_{t-ks}, \dots, x_{t-s}, \dots, x_{t-m}, \dots, x_t]$	$[x_{t+1}, \dots, x_{t+f}]$
$\vdots$	$\vdots$	$\vdots$

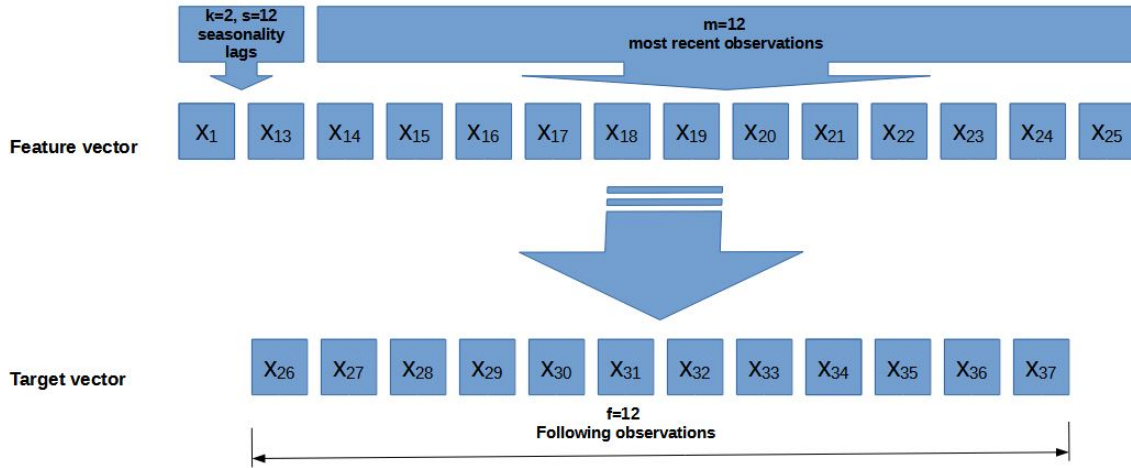


Fig. 6.2 The first feature vector $\mathbf{X}$ and target values $\mathbf{Y}$ of an SLSTM network with $s = 12$, $k = 2$, $m = 11$, and $f = 12$.

The available meteorological data, comprising time series from multiple stations, can be structured as **time-series panel data**. When applying the SLSTM model to such a dataset, two primary modeling strategies can be considered:

- **Univariate Approach:** In this strategy, a separate SLSTM model is trained for each individual meteorological station. This approach treats each time series independently, allowing the model to capture the specific dynamics of that single location.
- **Multivariate Approach:** This strategy involves training a single, unified SLSTM model on the entire panel dataset at once. By treating the observations from all stations as multiple input series, this approach has the potential to learn shared patterns and dependencies across different locations.

6.3.3 Univariate Approach: Implementation Details

In the univariate approach, an independent SLSTM model is trained for each meteorological station. The dataset for each station is partitioned into training and testing sets based on a temporal split. Specifically, the final r observations of the time series are held out as the test set, while the preceding observations are used for training. The choice of r is determined by the forecasting horizon of the analysis; for example, to predict the next 12 monthly records, one would set $r = 12$. Prior to training, it is standard practice to rescale each time series (or the imputed version thereof) to the range $[0, 1]$. This normalization prevents large input values from slowing down or destabilizing the network's learning process.

6.3.4 Multivariate Approach: Implementation Details

In the multivariate approach, a single, unified SLSTM model is trained on the entire panel of time series data. The dataset is partitioned by randomly selecting a subset of stations for training and holding out the remainder for testing. A common split is to use 80% of the stations for the training set and the remaining 20% for the test set.

A key advantage of this approach is the ability to incorporate static, station-specific features as exogenous regressors. Characteristics such as altitude and climate zone can be included as inputs to the model. This is achieved by augmenting the feature vector $\mathbf{X}$ (as described in Table 6.1 and Figure 6.2) with these additional regressors. As with the univariate approach, all input features—both the time series values and the exogenous regressors—must be rescaled to the range $[0, 1]$.

Furthermore, the multivariate framework lends itself to more robust validation techniques. Instead of relying on a single, arbitrary partition of stations, one can employ **stratified repeated random subsampling** or **stratified bootstrap resampling**. By repeatedly partitioning the dataset into training and testing sets (e.g., at an 80/20 ratio) many times, we can generate multiple predictions for each station in the test set. This ensemble of forecasts can then be used to compute a more stable point estimate (e.g., by averaging) and to construct empirical prediction intervals, providing a measure of forecast uncertainty.

6.4 Results

The proposed methodology was applied to a dataset of monthly precipitation time series from meteorological stations across the Peloponnese and the Ionian islands in

Greece. The data spans the period from 2010 to 2020. The primary objective of the analysis was to predict the annual precipitation levels for the following year at each station.

The data preprocessing and modeling pipeline was implemented using a combination of software tools. The two-step imputation process was conducted in **R**. Specifically, the **geosphere** package was used to calculate the Haversine distances for the spatial imputation step, and the **imputeTS** package was used to perform the temporal imputation via Kalman smoothing.

The SLSTM models were developed and trained in **Python** using the **Keras** API, with **TensorFlow** as the computational backend. The analysis was executed on a workstation with the following specifications:

- **CPU:** Intel® Core™ processor with a 4.80 GHz clock speed, 8 cores, and 16 threads.
- **Memory:** 79.9 GB of DDR4 RAM.
- **GPU:** NVIDIA GeForce RTX 3060 Ti with 8 GB of GDDR6 dedicated memory.

The SLSTM networks were configured using standard and recommended practices. For weight initialization, the **Xavier (or Glorot) initializer** was employed. This method initializes the weights of each layer by drawing from a zero-mean normal distribution with a specific variance, as shown below:

$$W_{i,j}^{[\ell]} \sim \mathcal{N}\left(0, \frac{1}{n^{[\ell-1]}}\right) \quad (6.1)$$

Here, $W_{i,j}^{[\ell]}$ is a weight in the ℓ -th layer matrix, and $n^{[\ell-1]}$ is the number of neurons (fan-in) in the preceding layer ($\ell - 1$). This initialization scheme is designed to maintain variance across layers, which helps to prevent the gradient from vanishing or exploding during training.

6.4.1 Data Description

The data for this study were sourced from the Greek automated weather stations network, which is operated by the Institute for Environmental Research and Sustainable Development of the National Observatory of Athens. This network began operations in 2007. From this network, a total of 64 stations located in the Peloponnese and the Ionian islands were selected for analysis. The earliest of these stations was commissioned in 2010.

For each of the 64 stations (their locations are depicted in Figure 6.3), the following data were utilized:

- **Time Series Data:** Monthly total precipitation records (in millimeters) for the duration of each station's operation.
- **Static Metadata:**
 - Geographical coordinates (latitude and longitude, in degrees).
 - Altitude (in meters).
 - Climate zone classification, as assigned based on the study by [288]. This feature is critical for capturing the distinct seasonality and memory patterns inherent in the precipitation data of different sub-regions.

6.4.2 Data Preprocessing

Data Cleaning

The initial dataset consisted of monthly precipitation time series for 64 stations, downloaded from the National Observatory of Athens. A multi-stage filtering process was applied to ensure data quality and suitability for time series analysis.

The filtering was performed as follows:

1. **Minimum Record Length:** Stations with records covering less than four years (not necessarily continuous) were removed. This step eliminated 11 stations, which were identified as being primarily for testing, temporary deployment, or were relatively new to the network. Eight of these stations were also noted to be non-operational in the final year of the study period.
2. **Extended Minimum Record Length:** An additional nine stations with operational periods of less than five years were also removed from the analysis to ensure all series were of sufficient length for robust modeling.
3. **Missing Value Threshold:** For the remaining 46 stations, the percentage of missing values within their respective operational periods was calculated. A threshold of 10% was established, and any station exceeding this limit was excluded. This step removed six additional stations, with the highest percentage of missing data being 18.27% at the Pinia station (the distribution of missing values is visualized in Figure 6.4).

After this rigorous filtering process, a final dataset comprising 40 high-quality meteorological stations was retained for the subsequent analysis.

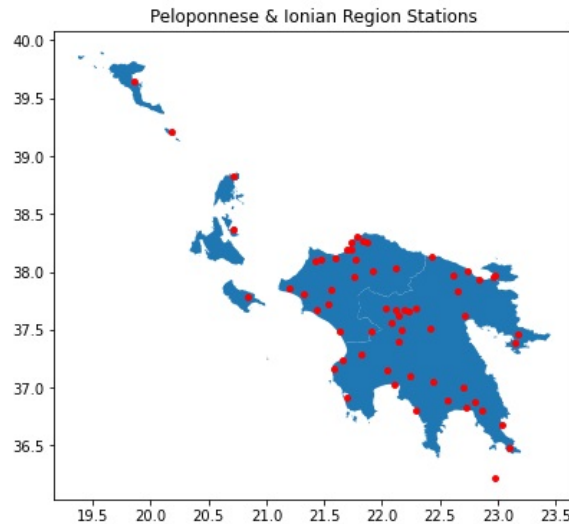


Fig. 6.3 The Peloponnese and the Ionian islands meteorological stations network (denoted with red dots) of the Institute for Environmental Research and Sustainable Development of the National Observatory of Athens.

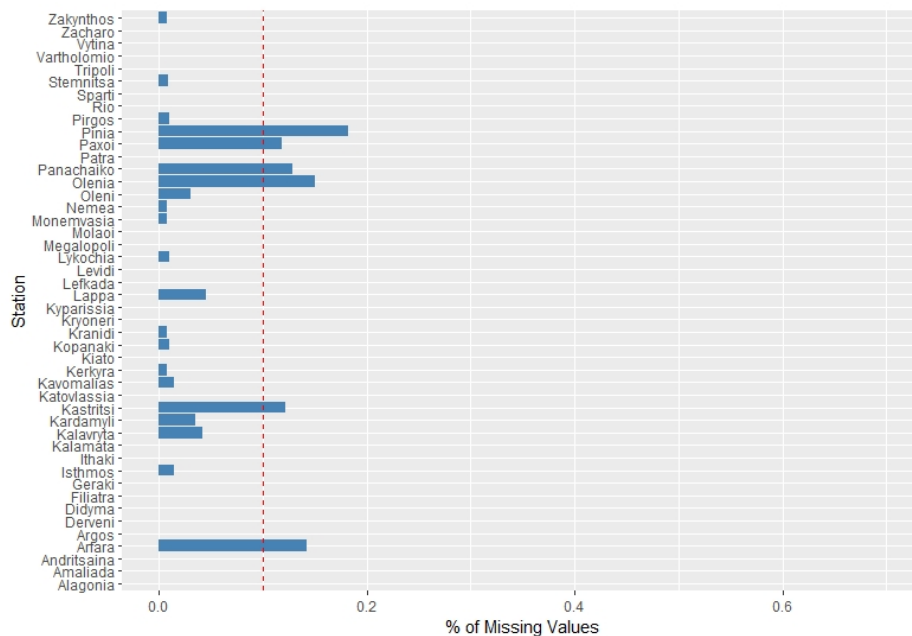


Fig. 6.4 Percentage of missing values in the meteorological stations that were in operation for at least 4 years during the period under study and were active in 2020. The red dotted line facilitates the identification of the stations with more than 10% of missing values during their operation period.

Data Imputation

The initial step in the imputation process involved assigning each of the 40 selected stations to a specific climate cluster. This assignment was based on the four distinct clusters identified from precipitation characteristics in the study by [288], which covers the same geographical area. For any meteorological station in our dataset not explicitly included in the aforementioned paper, assignment to a cluster was performed using the Thiessen polygons provided in the same study.

With stations grouped into climatologically coherent clusters, the spatial imputation was performed. First, the Haversine distances between all pairs of stations within each cluster were calculated. For each station, this provided a ranked list of its neighbors, ordered by proximity. The imputation algorithm then proceeded as follows: for any missing monthly value at a given station, the algorithm sought to replace it with the corresponding value from the closest station within the same cluster. If that value was also missing, the procedure iterated to the next-closest station in the ranked list. This process continued until a non-missing value was found. To prevent the use of data from geographically dissimilar locations, the search was terminated and the value left unimputed if the distance to the potential donor station exceeded 30 km.

To illustrate the principle behind this spatial imputation, Figure 6.5 presents the precipitation time series for two nearby stations, Megalopoli and Lykochia, which are separated by a Haversine distance of 10.64 km. The plot clearly shows a strong similarity in their precipitation patterns. The embedded plot, which provides a detailed view of the year 2018, highlights a missing value for the Lykochia station in May 2018. The congruent behavior of the two series provides a strong justification for using the value from the Megalopoli station to impute the missing record, validating the approach described in Subsection 6.3.1.

The spatial imputation procedure successfully filled the vast majority of missing values. Following this step, only a small number of observations remained missing across five of the time series (specifically, the Zakynthos, Kalavryta, Kranidi, Lappa, and Oleni stations), with 1, 4, 1, 6, and 4 missing observations, respectively. To address these final gaps, a temporal imputation method was applied. For each of the five series, the data were first log-transformed to stabilize variance and reduce skewness. A basic structural model, configured for monthly data (i.e., with a frequency of 12) and incorporating a local trend component, was then fitted to the log-transformed series. The missing values were subsequently imputed using Kalman smoothing based on this fitted model. This two-step spatial-then-temporal procedure resulted in a complete, gap-free panel dataset ready for predictive modeling.

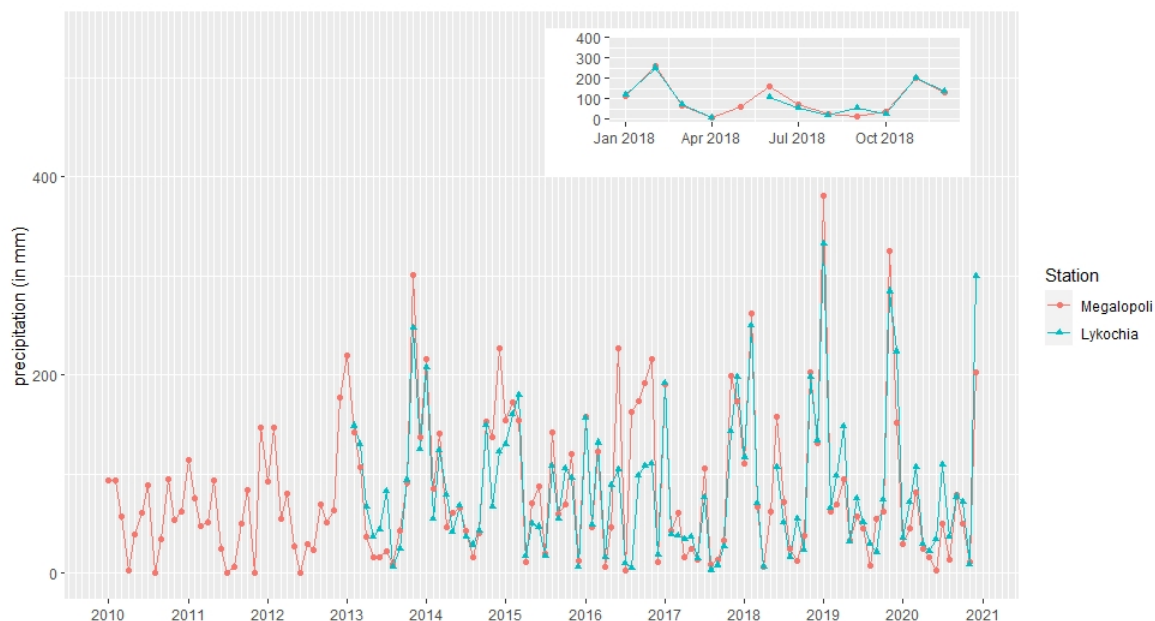


Fig. 6.5 Monthly precipitation time series plots for Megalopoli and Lykochia. In the embedded plot, the year 2018 is presented in detail.

6.4.3 LSTM Model Performance

Following the ML algorithm presented in Section 6.3.2, the SLSTM model was employed to predict precipitation at each station.

Model Architecture

A consistent five-layer neural network architecture was adopted for both the univariate and multivariate approaches to ensure a fair comparison. The architecture, depicted in Figure 6.6, is as follows:

1. An **input layer**, which receives the feature vectors.
2. An **LSTM layer**, which processes the temporal sequences.
3. A **hidden dense layer** with a sigmoid (logistic) activation function.
4. A second **hidden dense layer**, also using a sigmoid activation function.
5. An **output layer**, which produces the final forecast.

The same hyperparameters were used across both approaches.

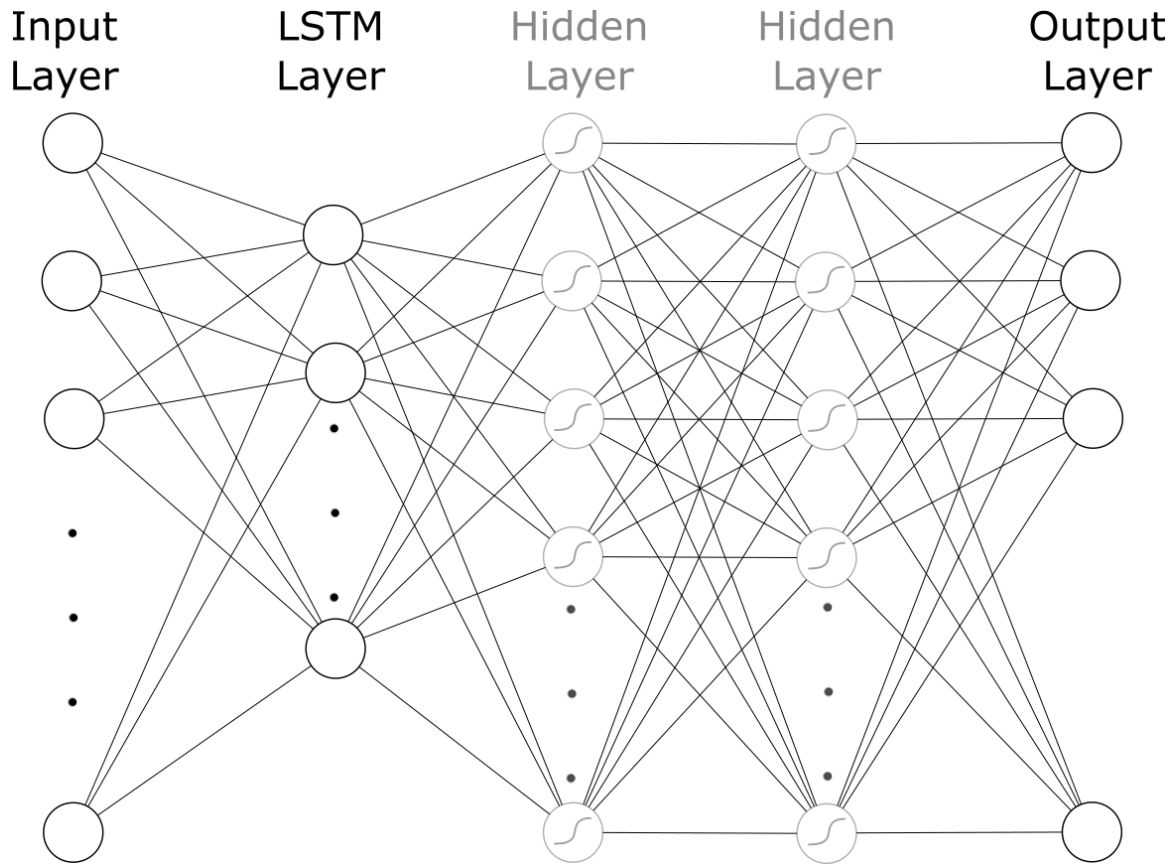


Fig. 6.6 The LSTM neural network architecture used in the analysis.

The construction of the feature vectors was designed to capture both recent trends and long-term seasonality, with the goal of predicting the total precipitation for the following year. Given the 12-month seasonality of precipitation, the following parameters were chosen for the SLSTM model: $s = 12$, $k = 2$, $m = 11$, and $f = 12$.

This parameterization means that to predict the next 12 months ($f = 12$), the model uses a feature vector composed of:

- The 12 most recent monthly observations (from $m = 0$ to $m = 11$).
- The corresponding monthly observations from the two preceding years (seasonal lags, $k = 2$).

For example, to predict the year following December 2020, the feature vector would include the 12 monthly values from 2020, as well as the precipitation values from December 2019 and December 2018.

For the **multivariate approach**, this feature vector was augmented with two static, station-specific exogenous regressors: the station's altitude and its assigned climate zone.

In all cases, a min-max normalization was applied to the log-transformed data before it was fed into the network.

Univariate Approach

For the univariate approach, the dataset was first partitioned into 40 distinct datasets, one for each meteorological station. A unified hyperparameter tuning strategy was then devised to ensure a consistent modeling approach across all stations.

The tuning process was conducted using the **Amaliada station** as a representative case. This station was selected because its time series length of 120 observations (10 years) represents the median operational duration among all 40 stations, making it a suitable proxy for the typical dataset size.

First, the Amaliada dataset was split into training and testing subsets. The final twelve feature vectors ($\mathbf{X}$) and their corresponding target values ($\mathbf{Y}$) were held out as the test set. These target values correspond to the 12 months of the final year of the record, 2020. Based on the SLSTM feature engineering parameters ($s = 12, k = 2, f = 12$), the total number of available input-output samples for the Amaliada station was $n - f - k \times s = 120 - 12 - 2 \times 12 = 84$, as described in Table 6.1.

A grid search was then performed over a predefined set of hyperparameter values to identify the optimal model configuration. To ensure the approach could be generalized to all stations (which have varying time series lengths, n), the search space for the hyperparameters was defined as a function of the data parameters n, f, k , and s . Specifically, the number of nodes (nd_ℓ) tested for the LSTM layer was selected from the following values:

$$nd_\ell = \begin{cases} \left\lfloor \frac{1}{10}(n - f - k \times s) \right\rfloor & (\text{for Amaliada station} = 7) \\ \left\lfloor \frac{1}{3}(n - f - k \times s) \right\rfloor & (\text{for Amaliada station} = 24) \\ n - f - k \times s & (\text{for Amaliada station} = 72) \end{cases}$$

while the number of nodes (nd_h) for each of the two hidden dense layers was tested from among the following values:

$$nd_h = \begin{cases} \left\lfloor \frac{1}{2}(n - f - k \times s) \right\rfloor & (\text{for Amaliada station} = 36) \\ 2(n - f - k \times s) & (\text{for Amaliada station} = 144) \\ 3(n - f - k \times s) & (\text{for Amaliada station} = 216) \end{cases}$$

The search space for the number of nodes in the LSTM layer (nd_ℓ) and the hidden dense layers (nd_h) was defined as follows, where $\lfloor \cdot \rfloor$ denotes the integer part:

- **LSTM Layer Nodes (nd_ℓ):**

$$\left\{ \frac{\lfloor n - f - k \times s \rfloor}{2}, \lfloor n - f - k \times s \rfloor \right\}$$

- **Hidden Layer Nodes (nd_h):**

$$\left\{ \frac{\lfloor n - f - k \times s \rfloor}{4}, \frac{\lfloor n - f - k \times s \rfloor}{2}, \lfloor n - f - k \times s \rfloor \right\}$$

These values were chosen to ensure that the number of nodes in the LSTM layer is no larger than the number of available input samples, while allowing the hidden layers to explore a wider range of complexities.

For the **batch size**, values of 1, 6, and 12 were tested. These choices reflect the temporal nature of the data, corresponding to a single month, a semester, and a full year, respectively. Across all experimental scenarios, the **Adam optimizer** was used for training.

The optimal combination of hyperparameters—batch size, number of LSTM nodes, and number of hidden layer nodes—was determined by selecting the configuration that minimized the **Mean Square Error (MSE)** loss function on the test set.

$$MSE = \frac{1}{n} \sum_i (Y_{ia} - Y_{ip})^2 \quad (6.2)$$

The optimal hyperparameter set for the univariate models was identified by selecting the configuration that minimized the MSE loss function, which is defined as:

$$MSE = \frac{1}{f} \sum_{i=1}^f (Y_{ia} - Y_{ip})^2 \quad (6.3)$$

where Y_{ia} and Y_{ip} are the actual and predicted values, respectively, for each of the f target time steps.

The results of the grid search, conducted on the Amaliada station data, are presented in Table 6.2. The best-performing model achieved an MSE of **0.02660**. This optimal configuration, marked in bold in the table, corresponds to the following hyperparameters:

- **Batch Size: 1**

- **LSTM Layer Nodes:** 7
- **Hidden Layer Nodes:** 36 for both dense layers

This set of hyperparameters was subsequently adopted for all 40 univariate station models to generate the final forecasts presented in Subsection 6.4.4. It is worth noting that a supplementary analysis using the Mean Absolute Percentage Error (MAPE) as the evaluation metric yielded the same optimal hyperparameter combination, confirming the robustness of this choice. The computational costs associated with training this selected model on the Amaliada station data are reported in Table 6.3.

Table 6.2 Grid search for the univariate approach using the Amaliada station. The smallest MSE value is marked in bold.

Batch Size	LSTM Nodes	Hidden Nodes	MSE
1	7	36	0.02660
		144	0.02805
		216	0.02889
	24	36	0.02761
		144	0.02815
		216	0.02798
	72	36	0.02827
		144	0.02820
		216	0.02747
6	7	36	0.02705
		144	0.02779
		216	0.02761
	24	36	0.02753
		144	0.02787
		216	0.02871
	72	36	0.02697
		144	0.02741
		216	0.02803
12	7	36	0.02764
		144	0.02716
		216	0.02756
	24	36	0.02684
		144	0.02775
		216	0.02776
	72	36	0.02717
		144	0.02790
		216	0.02762

Table 6.3 The CPU usage range and the time required to run the analysis for the selected hyperparameters.

	CPU Usage	Time (in Sec)
Univariate	12–16%	106.84
Multivariate	17–21%	314.11

Multivariate Approach

For the multivariate approach, a different strategy for data partitioning and hyperparameter tuning was required. The dataset of 40 stations was first split by randomly selecting 32 stations (80%) for the training set and holding out the remaining 8 stations (20%) for the test set. This initial partition was used to guide the hyperparameter selection process.

The feature vector $\mathbf{X}$ for the multivariate model was constructed as previously described, but was augmented with the two available static exogenous regressors for each station: altitude and climate zone.

Given the larger and more complex input space of the multivariate model, a common heuristic was adopted to constrain the model’s complexity and mitigate the risk of overfitting. This “rule of thumb” sets the number of neurons in the hidden layers based on the size of the training data and the input/output dimensions. Specifically, the number of neurons in the hidden layers was set as follows:

$$N_h = \frac{N_s}{\alpha \times (N_i \times N_o)} \quad (6.4)$$

$$N_h = \frac{N_s}{\alpha \cdot (N_i + N_o)} \quad (6.5)$$

where N_i is the number of input neurons, N_o is the number of output neurons, and N_s is the number of samples in the training dataset. The parameter α is a scaling factor, typically between 2 and 10. For this analysis, the search space for α was set to $\{2, 5, 10\}$, resulting in three different potential sizes for the hidden layers.

For the number of nodes in the LSTM layer, the values tested were $\frac{n-f-k \times s}{10}$, $\frac{n-f-k \times s}{20}$, and $\frac{n-f-k \times s}{30}$, where n is the total number of observations in the training set.

Finally, consistent with the univariate case, the batch size was tested with values of 1, 6, and 12. The Adam optimizer and the Mean Square Error (MSE) loss function were used to guide the selection of the best hyperparameter set.

The results of this grid search are presented in Table 6.4. The optimal configuration, which achieved the lowest MSE of **0.01576**, is marked in bold. This corresponds to:

- **Batch Size:** 12
- **LSTM Layer Nodes:** $\frac{n-f-k \times s}{30}$
- **Hidden Layer Nodes (N_h):** Calculated using Equation (6.5) with $\alpha = 10$.

Table 6.4 Grid search for the multivariate approach using the initial split. The smallest *MSE* value is marked in bold.

Batch Size	LSTM Nodes	Hidden Nodes	<i>MSE</i>
1	98	140	0.01722
		280	0.01739
		701	0.01868
	148	140	0.01686
		280	0.01724
		701	0.01864
	296	140	0.01678
		280	0.01668
		701	0.01895
6	98	140	0.01643
		280	0.01755
		701	0.01825
	148	140	0.01599
		280	0.01685
		701	0.01883
	296	140	0.01674
		280	0.01650
		701	0.01839
12	98	140	0.01576
		280	0.01695
		701	0.01900
	148	140	0.01591
		280	0.01642
		701	0.01855
	296	140	0.01560
		280	0.01587
		701	0.01806

The optimal hyperparameter configuration, identified in the initial tuning phase, was subsequently employed in a more extensive analytical procedure to generate robust

forecasts. Specifically, a resampling method involving 1000 independent, random partitions of the data was performed. In each of the 1000 iterations, the dataset was randomly split into a training set (32 stations) and a test set (8 stations), and the optimized model was trained and validated. This process resulted in an ensemble of 1000 distinct one-year-ahead forecasts for the stations in the test sets. The analysis of these predictions is presented in Subsection 6.4.4.

Regarding computational performance, the resource utilization and processing time for a single training run with the selected hyperparameter configuration are documented in Table 6.3. It is important to note that for the full multivariate analysis involving 1000 iterations, the total computational time was directly proportional to the number of iterations. Additionally, a positive correlation was observed between processing time and the number of nodes in both the LSTM and hidden layers.

6.4.4 Predicting

Both the univariate and multivariate methodologies were deployed to forecast the monthly precipitation for the year 2021. The input data for this forecasting task were based on the information available as of the end of December 2020, which, according to the feature engineering scheme, comprised the 12 monthly measurements from 2020 and the December measurements from 2019 and 2018.

To visually compare the performance and characteristics of the two approaches, Figure 6.7 presents the results for two illustrative stations: Monemvasia (left plot) and Alagonia (right plot). The elements of the figure are as follows:

- **Multivariate Forecasts:** The boxplots represent the distribution of the 1000 forecasts generated for each month by the multivariate methodology. These forecasts were back-transformed from the logarithmic scale to their original scale (mm of precipitation). The boxplots effectively visualize the forecast uncertainty and highlight the months projected to have low or high precipitation.
- **Univariate Forecasts:** The point forecasts generated by the single univariate model for each station are indicated by blue circles.
- **Actual Values:** The true measured precipitation values for each month in 2021 are indicated by orange circles, serving as the ground truth for comparison.

This visualization provides a clear comparison of the point forecasts from the univariate models against the forecast distributions produced by the more complex multivariate resampling approach.

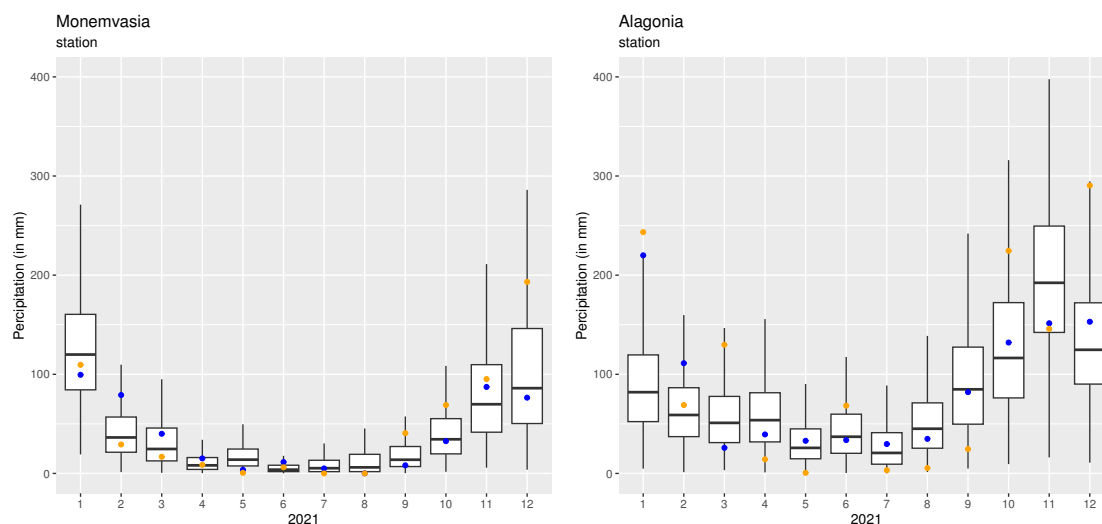


Fig. 6.7 The boxplots of the predictions for each month for the Monemvasia (left plot) and Alagonia (right plot) stations based on the multivariate approach. The corresponding predictions based on the univariate approach are denoted with a blue circle. The true measured values for the two stations for each month are denoted with orange circles.

To provide a comparative benchmark for the proposed SLSTM models, a classical time series forecasting method, the **Seasonal Autoregressive Integrated Moving Average (SARIMA)** model, was also implemented. For the two illustrative stations (Monemvasia and Alagonia), an optimal SARIMA model was identified for each time series using the `auto.arima` function from the `forecast` package in R, which automates the model selection process based on information criteria.

Table 6.5 presents a detailed comparison of the monthly forecasts for 2021. It includes:

- The forecasts from the proposed SLSTM algorithm (specifically, the mean or point forecast from the multivariate approach).
- The 12-month-ahead predictions generated by each station's optimal SARIMA model.
- The actual measured precipitation values (in mm).
- The absolute deviation between each forecast and the actual value, shown in parentheses. The smallest absolute deviation for each month is highlighted in bold.

A review of the values in Table 6.5 indicates that the two methods offer balanced performance when considering the absolute errors across all months for both stations. However, a notable trend is that the proposed SLSTM method tends to underestimate the total annual precipitation for both stations.

While a simple comparison of absolute errors shows a balanced performance, a deeper look at the error distribution reveals a key advantage of the proposed SLSTM method. For the total annual precipitation, the SLSTM model's absolute percentage error demonstrates greater stability across different climate zones (19.43% for Monemvasia vs. 14.23% for Alagonia). In contrast, the SARIMA models are less stable, exhibiting a low error of 7.52% for the drier Monemvasia station but a much larger error of 40.95% for the rainier Alagonia station.

Further insight into the proposed method's capabilities is provided in Figure 6.8, which visualizes the **kernel density estimates** of the total annual precipitation forecasts. These densities were generated from the ensemble of 1000 annual predictions for each station, where each annual prediction is the sum of the 12 forecasted monthly values. In the figure, the stations are ordered by climate zone, from driest to rainiest, with brackets indicating stations belonging to the same climate group as defined by [288]. The white vertical lines within each density plot mark the first and third quartiles, and the blue dots represent the actual observed total precipitation for 2021.

This analysis highlights two significant strengths of the proposed prediction algorithm:

1. **Climate Zone Differentiation:** The algorithm successfully captures the distinct characteristics of the different climate zones. For example, the forecast distributions for the Alagonia and Stemnitsa stations, both in high-altitude, rainy areas of the western Peloponnese, correctly reflect their high-precipitation climate. Conversely, the predictions for the Isthmos and Didyma stations accurately identify them as belonging to the driest region of the study area.
2. **Probabilistic Forecasting:** The kernel density estimates and their corresponding quartiles provide a natural way to construct credible intervals for the annual precipitation forecasts. This probabilistic information is invaluable for studying a range of realistic precipitation scenarios, moving beyond simple point predictions.

Table 6.5 The monthly actual precipitation values (in mm) and the predictions using the recently proposed algorithm and the prediction for 12 months ahead based on the best SARIMA model. In the parentheses, the absolute differences between each prediction from the actual value are reported. The minimum for each month and each station are marked in bold.

Month	Monemvasia Station			Alogonia Station		
	Predictions			Predictions		
	Actual	Univariate	SARIMA *	Actual	Univariate	SARIMA **
January	109.6	99.5 (10.1)	120.3 (10.7)	243.4	220.0 (23.4)	78.4 (165.0)
February	29.2	79.1 (49.9)	59.3 (30.1)	69.0	111.2 (42.2)	70.0 (1.0)
March	16.8	40.0 (23.2)	54.1 (37.3)	129.8	25.8 (104.0)	72.8 (57.0)
April	8.7	15.1 (6.4)	11.2 (2.5)	14.2	39.3 (25.1)	64.7 (50.5)
May	0.6	3.7 (3.1)	6.5 (5.9)	0.6	32.9 (32.3)	27.4 (26.8)
June	6.2	11.5 (5.3)	8.9 (2.7)	68.4	33.6 (34.8)	6.6 (61.8)
July	0.0	5.0 (5.0)	0.9 (0.9)	3.2	29.6 (26.4)	35.0 (31.8)
August	0.0	0.1 (0.1)	0.2 (0.2)	5.4	34.9 (29.5)	24.8 (19.4)
September	40.6	8.2 (32.4)	21.9 (18.7)	24.6	82.0 (57.4)	68.8 (44.2)
October	69.0	32.6 (36.4)	71.9 (2.9)	224.6	132.0 (92.6)	60.5 (164.1)
November	95.2	87.3 (7.9)	117.8 (22.6)	145.8	151.5 (5.7)	66.7 (79.1)
December	193.2	76.4 (116.8)	138.9 (54.3)	290.4	153.0 (137.4)	144.3 (146.1)
Total (annual)	569.1	458.5	611.9	1219.4	1045.8	720.0

SARIMA(0,0,0)*x*(2,0,0)₁₂, ** *SARIMA*(1,0,0)*x*(0,1,1)₁₂.

*

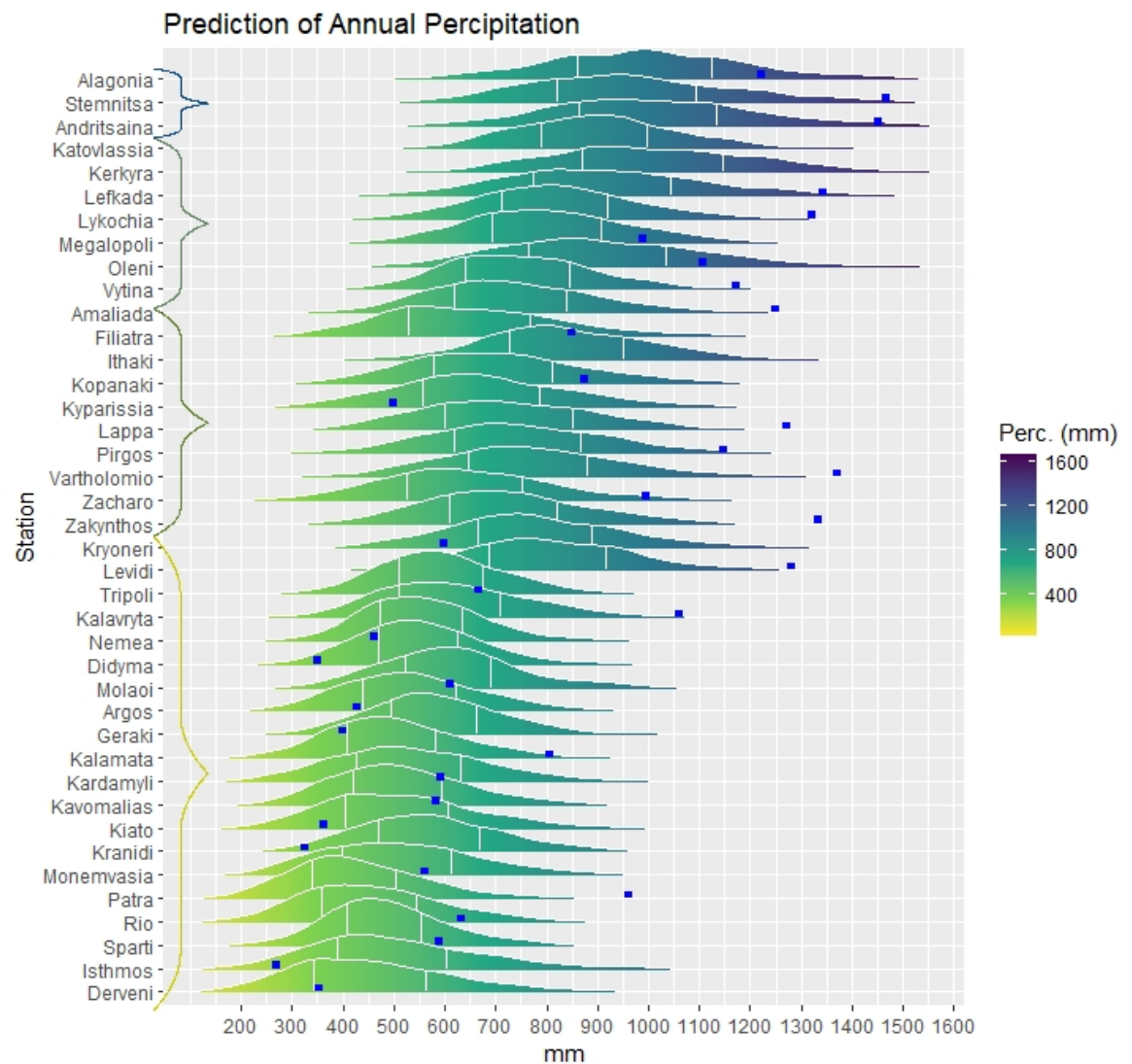


Fig. 6.8 The kernel density estimates for the annual total precipitation (sum of 12 months' predictions) for all stations. The clusters of the stations according to their climate zone are denoted with brackets. The white lines in the density plots indicate the first and the third quartile. The blue dots represent the observed values of the total precipitation for each station in 2021.

Despite the overall success, it is evident that for certain stations, such as Zakynthos and Patra, the model failed to accurately estimate the total precipitation. This underestimation can likely be attributed to two primary factors:

1. **Anomalous Climate Conditions:** The year 2021 experienced a 36% average increase in precipitation across the study area compared to 2020. Such a substan-

tial and abrupt shift in climatic conditions presents a significant challenge for any predictive algorithm, as the training data does not contain a similar precedent.

2. **Systematic Under-Recording in Input Data:** Meteorological data collection is often imperfect. Brief maintenance periods, power outages, or sensor malfunctions can lead to incomplete monthly records. This frequently results in artificially low monthly precipitation totals in the historical data. When a model is trained on data with this systematic downward bias, it will naturally learn to produce forecasts that also underestimate future observations.

The tendency for the model to underestimate precipitation, which was observed in both the univariate and multivariate approaches, is a finding consistent with other research in this domain. For example, similar underestimation biases in ANN-based precipitation forecasting have been reported in studies conducted in Iran by [289] and in Korea by [290]. This suggests that the observed bias may be an inherent characteristic of applying this class of models to precipitation data, rather than a flaw specific to this implementation.

6.5 Conclusions

The accurate prediction of meteorological variables, such as precipitation, is of paramount importance for managing energy production, particularly in regions heavily reliant on renewable sources. In these contexts, hydropower serves as a critical stabilizing factor for the electricity grid. The algorithm presented in this chapter provides a robust framework for long-range precipitation forecasting—for example, a full year ahead—using data from an automated weather station network.

The proposed two-step imputation procedure offers a simple yet effective method for creating a complete dataset from sparse and heterogeneous raw data. Its primary complexity lies in the initial computation of Haversine distances and the assignment of stations to climatologically coherent zones. Following this preprocessing, the two modeling approaches presented—univariate and multivariate—provide the necessary tools for generating forecasts. While the multivariate approach offers more comprehensive and robust insights, particularly through its ability to quantify uncertainty, the univariate approach remains a valuable tool for generating point estimates suitable for baseline energy production scenarios. The primary challenge, as with any ANN-based methodology, lies in managing model complexity and selecting appropriate hyperpa-

rameters. However, this study has demonstrated that standard architectures combined with a principled grid search can yield reliable solutions.

This research represents a foundational step toward applying advanced ML techniques for annual rainfall prediction. Future work should focus on exploring more diverse network architectures and alternative models to further improve forecast accuracy. Moreover, a more in-depth comparative study is needed to systematically identify the conditions under which the univariate or multivariate approach is superior. Finally, we hope this study provides a solid basis for future research into the scalability of this methodology to larger temporal and spatial datasets and for sensitivity analyses on different architectural choices, activation functions, and other hyperparameter configurations.

6.6 Summary

This chapter presents a ML framework to address the critical need for accurate, long-range precipitation forecasting, essential for managing hydropower operations and stabilizing the electricity grid in regions with high renewable energy penetration. Focusing on Greece’s Peloponnese and Ionian islands—a region representing 21% of the nation’s renewable capacity—the study develops a methodology to support energy security in the context of the EU’s Renewable Energy Directive 2018/2001/EU.

The methodology is designed to handle sparse and incomplete meteorological data from a network of 40 automated weather stations. It features a novel two-step imputation process that first uses spatial information (Haversine distance within climate zones) to fill gaps, followed by temporal imputation using structural models and Kalman smoothing. The core of the prediction framework is a SLSTM network, architected to capture both short-term dependencies and long-term seasonal cycles. This model is evaluated in two distinct frameworks: a **univariate** approach, modeling each station independently, and a **multivariate** approach, which trains a single model on the entire panel of stations, incorporates exogenous variables (altitude, climate zone), and uses a resampling strategy to generate probabilistic forecasts.

Results for the 2021 forecast horizon demonstrate the superiority of the multivariate SLSTM (MSE=0.01576) over the univariate approach (MSE=0.02660). When benchmarked against a classical SARIMA model, the SLSTM framework provides more stable error performance across diverse climate zones (14–19% error vs. 7–41% for SARIMA). The probabilistic forecasts from the multivariate model successfully differentiate regional climate characteristics and provide valuable uncertainty quantification

through kernel density estimates. A systematic underestimation tendency is noted, a phenomenon consistent with related literature and attributed to anomalous climate conditions in the test year and potential data quality issues.

In conclusion, this study develops and validates an end-to-end framework for precipitation forecasting that provides not only accurate point estimates but also robust uncertainty quantification. The methodology serves as a valuable tool for optimizing hydropower operations and enhancing energy security in renewable-dominant systems, providing a foundation for future research in model architecture and scalability.

Chapter 7

Closing Remarks

7.1 Discussion

This dissertation has examined the challenge of real-time monitoring in high-dimensional, nonstationary data streams through the integration of Statistical Process Monitoring and Machine Learning. Its main contribution is the development and evaluation of the Skape (SK) family of cluster validity indices, which serve as novel monitoring statistics designed to overcome limitations of classical multivariate control charts.

The methodological core presented in Chapters 3 and 4 established that clustering-based separability metrics provide a robust and scalable monitoring framework for high-dimensional data streams. The SK index family, particularly within the Consecutive Sample (CONS) framework, demonstrated strong detection performance under high-dimensional conditions where traditional methods degraded notably. Since the framework requires only pairwise distance computations, it avoids the inversion of covariance matrices, offering practical advantages for real-time deployment in resource-constrained settings. Four significant findings emerged: (i) graceful degradation of performance in high dimensions compared to classical methods, (ii) robustness across different distributional environments, (iii) reliable detection of both location and scale shifts, and (iv) computational efficiency suited for stream monitoring.

Chapters 5 and 6 showed the adaptability of machine learning techniques across diverse application domains, suggesting future potential for unified and automated data-driven monitoring systems. When integrated with the SK-based core, these applications move toward end-to-end frameworks encompassing preprocessing, anomaly detection, predictive analytics, and decision support.

Despite its strengths, the framework has several important limitations. The SK indices rely on an initial assumption of a known or estimable 'in-control' state, generally

achieved through simulation or semi-supervised approaches. While this provides a justified statistical basis for setting thresholds under distributional assumptions, general results for arbitrary distributions remain limited. The dependence on cluster structure assumes detectable groupings representing normal operating conditions; this is suitable for structured settings but less robust for unstructured or continuously evolving data.

Scalability is another practical concern. Although more efficient than covariance-based methods, performance under extremely high-dimensional and high-velocity conditions may require approximation methods or hardware optimization. Additionally, since data are processed in subgroups and clusters recomputed incrementally, extending the approach to fully online, adaptive modes of operation remains a promising but incomplete goal.

Most studies were performed with retrospective datasets to enable controlled validation. Testing under real-time constraints involving latency, resource efficiency, and integration in operational systems is therefore a future priority. Broader validation across domains and conditions will be essential to demonstrate performance consistency and domain transferability.

7.2 Future Horizons

The results in this dissertation outline a foundation for continued development of intelligent, adaptive monitoring systems while revealing several directions for further improvement. Future research should focus on deriving distribution-independent control limits to improve applicability across heterogeneous data environments. Establishing theoretical guarantees for control limits under minimal assumptions would broaden the operational reliability of SK-based frameworks.

Addressing dependence on cluster formation is another open challenge. Automated determination of cluster relevance, dynamic cluster adaptation, and online clustering mechanisms could increase robustness under changing data structures. Developing strategies that handle non-clustered operational data, such as density-based or manifold-based representations, may further generalize the approach. Enhancing scalability will also be crucial. Future work may incorporate approximate nearest-neighbor search, parallel computation, or low-rank distance representations to ensure responsiveness under real-time processing demands. The current semi-offline nature of the analysis invites development of continuously updating variants capable of adjusting control limits on the fly. Adopting adaptive thresholds and recursive similarity calculations could help transition the framework toward true streaming analytics.

Finally, deeper operational testing in real-world settings across domains like manufacturing, environmental systems, and maritime operations should be pursued. These studies should examine memory footprint, update costs, and integration into decision-support pipelines. Incorporating predictive modeling components could eventually enable proactive maintenance and early-warning systems linked directly to monitored process stability metrics.

In sum, while this dissertation has delivered methods that significantly improve existing monitoring capabilities, their full deployment potential depends on addressing current limitations regarding distributional assumptions, data structure variability, real-time adaptation, and large-scale system integration. Exploring these practical directions will enable the development of more adaptive, efficient, and autonomous monitoring systems capable of meeting future demands.

Bibliography

- [1] Mohamed Medhat Gaber, Arkady Zaslavsky, and Shonali Krishnaswamy. “A survey of classification methods in data streams”. In: *Data Streams: Models and Algorithms* (2007), pp. 39–59.
- [2] Tianyuan Lu, Lei Wang, and Xiaoyong Zhao. “Review of anomaly detection algorithms for data streams”. In: *Applied Sciences* 13.10 (2023), p. 6353.
- [3] Ibrahim Abaker Targio Hashem et al. “The rise of “big data” on cloud computing: Review and open research issues”. In: *Information systems* 47 (2015), pp. 98–115.
- [4] Varun Chandola, Arindam Banerjee, and Vipin Kumar. “Anomaly detection: A survey”. In: *ACM computing surveys (CSUR)* 41.3 (2009), pp. 1–58.
- [5] Robertas Damaševičius, Nebojsa Bacanin, and Sanjay Misra. “From sensors to safety: Internet of Emergency Services (IoES) for emergency response and disaster management”. In: *Journal of Sensor and Actuator Networks* 12.3 (2023), p. 41.
- [6] Annie Ibrahim Rana et al. “Anomaly detection guidelines for data streams in big data”. In: *2016 3rd International Conference on Soft Computing & Machine Intelligence (ISCMCI)*. IEEE. 2016, pp. 94–98.
- [7] Hui Lü et al. “Non-Probabilistic Uncertainty and Correlation Propagation Analysis Methods Based on Multidimensional Parallelepiped Model.” In: *International Journal of Computational Methods* 21.2 (2024).
- [8] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. “Efficient algorithms for mining outliers from large data sets”. In: *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. 2000, pp. 427–438.
- [9] Peter J Rousseeuw. “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis”. In: *Journal of computational and applied mathematics* 20 (1987), pp. 53–65.
- [10] Joseph C Dunn. “Well-separated clusters and optimal fuzzy partitions”. In: *Journal of cybernetics* 4.1 (1974), pp. 95–104.
- [11] Xueqin Zhang, Chunhua Gu, and Jiajun Lin. “Support vector machines for anomaly detection”. In: *2006 6th World Congress on Intelligent Control and Automation*. Vol. 1. IEEE. 2006, pp. 2594–2598.
- [12] Julien Lesouple et al. “Generalized isolation forest for anomaly detection”. In: *Pattern Recognition Letters* 149 (2021), pp. 109–119.

- [13] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. “Isolation forest”. In: *2008 eighth IEEE international conference on data mining*. IEEE. 2008, pp. 413–422.
- [14] Finale Doshi-Velez and Been Kim. “Towards a rigorous science of interpretable machine learning”. In: *arXiv preprint arXiv:1702.08608* (2017).
- [15] Diogo V Carvalho, Eduardo M Pereira, and Jaime S Cardoso. “Machine learning interpretability: A survey on methods and metrics”. In: *Electronics* 8.8 (2019), p. 832.
- [16] Kyriakos Skarlatos et al. “Assessing Ships’ Environmental Performance Using Machine Learning”. In: *Energies* 16.6 (2023), p. 2544.
- [17] Kyriakos Skarlatos et al. “Ship engine model selection by applying machine learning classification techniques using imputation and dimensionality reduction”. In: *Journal of Marine Science and Engineering* 12.1 (2024), p. 97.
- [18] Kyriakos Skarlatos et al. “Projecting annual rainfall timeseries using machine learning techniques”. In: *Energies* 16.3 (2023), p. 1459.
- [19] Fouzi Harrou et al. *Statistical process monitoring using advanced data-driven and deep learning approaches: theory and practical applications*. Elsevier, 2020.
- [20] Joao Gama. *Knowledge Discovery from Data Streams*. CRC Press, 2010.
- [21] Pedro Domingos and Geoff Hulten. “Mining high-speed data streams”. In: *Proceedings of the sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. 2000, pp. 71–80.
- [22] Albert Bifet et al. “MOA: Massive Online Analysis”. In: *Journal of Machine Learning Research* 11 (2010), pp. 1601–1604.
- [23] Robin Sommer and Vern Paxson. “Outside the closed world: On using machine learning for network intrusion detection”. In: *IEEE Symposium on Security and Privacy* (2010), pp. 305–316.
- [24] Gari D Clifford, Francisco Azuaje, and Patrick E McSharpy. *Advanced Methods and Tools for ECG Data Analysis*. Artech House, 2006.
- [25] Edward A Lee. “Cyber physical systems: Design challenges”. In: *IEEE Design & Test* 26.1 (2008), pp. 9–18.
- [26] Siddhartha Bhattacharyya et al. “Data mining for credit card fraud: A comparative study”. In: *Decision Support Systems* 50.3 (2011), pp. 602–613.
- [27] João Gama et al. “A survey on concept drift adaptation”. In: *ACM computing surveys (CSUR)* 46.4 (2014), pp. 1–37.
- [28] Albert Bifet and Ricard Gavaldà. “Adaptive Hoeffding Tree”. In: *International Symposium on Intelligent Data Analysis*. Springer, 2009, pp. 36–50.
- [29] Jie Xu et al. “Edge learning: A survey on distributed machine learning at the edge”. In: *arXiv preprint arXiv:1909.11173* (2020).
- [30] Michel Basseville and Igor V Nikiforov. *Detection of Abrupt Changes: Theory and Application*. Prentice Hall, 1993.
- [31] Robert R Everett. “The whirlwind I computer”. In: *Papers and discussions presented at the Dec. 10-12, 1951, joint AIEE-IRE computer conference: Review of electronic digital computers*. 1951, pp. 70–74.

- [32] Hermann Kopetz. *Real-time systems: design principles for distributed embedded applications*. Springer, 1997.
- [33] Wayne L Winston. *Operations research: applications and algorithm*. Thomson Learning, Inc., 2004.
- [34] Warren B Powell. *Approximate Dynamic Programming: Solving the curses of dimensionality*. Vol. 703. John Wiley & Sons, 2007.
- [35] Gary Klein. “Developing expertise in decision making”. In: *Thinking & Reasoning* 3.4 (1997), pp. 337–352.
- [36] Rudolph Emil Kalman. “A new approach to linear filtering and prediction problems”. In: (1960).
- [37] Abraham Wald. *Sequential analysis*. Courier Corporation, 2004.
- [38] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.
- [39] Vincent D Blondel and Alexandre Megretski. *Unsolved problems in mathematical systems and control theory*. Princeton University Press, 2009.
- [40] Jane K Hart and Kirk Martinez. “Environmental sensor networks: A revolution in the earth system science?”. In: *Earth-Science Reviews* 78.3-4 (2006), pp. 177–191.
- [41] Douglas C Montgomery. *Introduction to Statistical Quality Control*. John Wiley & Sons, 2009.
- [42] Sotiris Bersimis, Stelios Psarakis, and John Panaretos. “Multivariate statistical process control charts: an overview”. In: *Quality and Reliability engineering international* 23.5 (2007), pp. 517–543.
- [43] Sotiris Bersimis, Aggeliki Sgora, and Stelios Psarakis. “The application of multivariate statistical process monitoring in non-industrial processes”. In: *Quality Technology & Quantitative Management* 15.4 (2018), pp. 526–549.
- [44] Walter Andrew Shewhart. “Economic quality control of manufactured product 1”. In: *Bell System Technical Journal* 9.2 (1930), pp. 364–389.
- [45] Douglas C Montgomery. *Design and analysis of experiments*. John wiley & sons, 2017.
- [46] Joseph M Juran. *Juran’s quality handbook*. 1999.
- [47] William H Woodall and Douglas C Montgomery. “Research issues and ideas in statistical process control”. In: *Journal of quality technology* 31.4 (1999), pp. 376–386.
- [48] Martin J Crowder and David J Hand. *Analysis of repeated measures*. Routledge, 1990.
- [49] Douglas M Hawkins. “Diagnostics for conformity of paired quantitative measurements”. In: *Statistics in Medicine* 21.13 (2002), pp. 1913–1935.
- [50] Zachary G Stoumbos and Marion R Reynolds Jr. “Economic statistical design of adaptive control schemes for monitoring the mean and variance: An application to analyzers”. In: *Nonlinear Analysis: Real World Applications* 6.5 (2005), pp. 817–844.

- [51] Robert S Kaplan. "Strategic performance measurement and management in nonprofit organizations". In: *Nonprofit management and Leadership* 11.3 (2001), pp. 353–370.
- [52] E. S. Page. "Continuous inspection schemes". In: *Biometrika* 41.1/2 (1954), pp. 100–115.
- [53] G. Barrie Wetherill and Don W. Brown. *Statistical Process Control: Theory and Practice*. Springer US, 2013. ISBN: 9781489929501.
- [54] Stuart W Roberts. "Control chart tests based on geometric moving averages". In: *Technometrics* 42.1 (2000), pp. 97–101.
- [55] Harry V Roberts. "Stock-market" patterns" and financial analysis: methodological suggestions". In: *The Journal of Finance* 14.1 (1959), pp. 1–10.
- [56] Douglas C Montgomery. *Introduction to statistical quality control*. John wiley & sons, 2020.
- [57] Cynthia A Lowry et al. "A multivariate exponentially weighted moving average control chart". In: *Technometrics* 34.1 (1992), pp. 46–53.
- [58] Harold Hotelling. "Multivariate quality control illustrated by air testing of sample bombsights". In: *Techniques of statistical analysis* (1947), p. 111.
- [59] Douglas M Hawkins. "Multivariate quality control based on regression-adjusted variables". In: *Technometrics* 33.1 (1991), pp. 61–75.
- [60] Douglas M Hawkins. "Regression adjustment for variables in multivariate quality control". In: *Journal of quality Technology* 25.3 (1993), pp. 170–182.
- [61] Karl Pearson. "LIII. On lines and planes of closest fit to systems of points in space". In: *The London, Edinburgh, and Dublin philosophical magazine and journal of science* 2.11 (1901), pp. 559–572.
- [62] Ian Jolliffe. "Principal component analysis". In: *Encyclopedia of statistics in behavioral science* (2005).
- [63] Manabu Kano et al. "A new multivariate statistical process monitoring method using principal component analysis". In: *Computers & chemical engineering* 25.7-8 (2001), pp. 1103–1113.
- [64] Bhavik R Bakshi. "Multiscale PCA with application to multivariate statistical process monitoring". In: *AIChE journal* 44.7 (1998), pp. 1596–1610.
- [65] Husnain Ali et al. "Multiscale principal component analysis-signed directed graph based process monitoring and fault diagnosis". In: *ACS omega* 7.11 (2022), pp. 9496–9512.
- [66] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. "Nonlinear component analysis as a kernel eigenvalue problem". In: *Neural computation* 10.5 (1998), pp. 1299–1319.
- [67] Zhichao Li et al. "Dynamic nonlinear process monitoring based on dynamic correlation variable selection and kernel principal component regression". In: *Journal of the Franklin Institute* 359.9 (2022), pp. 4513–4539.
- [68] Manish H Bharati and John F MacGregor. "Multivariate image analysis for real-time process monitoring and control". In: *Industrial & Engineering Chemistry Research* 37.12 (1998), pp. 4715–4724.

- [69] Sangyeol Lee and Youngmi Lee. “Sequential online monitoring for autoregressive time series of counts”. In: *Journal of the Korean Statistical Society* 53.2 (2024), pp. 307–327.
- [70] Salah Bouhouche et al. “Quality monitoring using principal component analysis and fuzzy logic. Application in continuous casting process”. In: *American Journal of Applied Sciences* 4.637-644 (2007), p. 24.
- [71] Svante Wold, Michael Sjöström, and Lennart Eriksson. “PLS-regression: a basic tool of chemometrics”. In: *Chemometrics and intelligent laboratory systems* 58.2 (2001), pp. 109–130.
- [72] Junghui Chen and Chien-Mao Liao. “Dynamic process fault monitoring based on neural network and PCA”. In: *Journal of Process control* 12.2 (2002), pp. 277–289.
- [73] Yingwei Zhang and Zhiyong Hu. “Multivariate process monitoring and analysis based on multi-scale KPLS”. In: *Chemical Engineering Research and Design* 89.12 (2011), pp. 2667–2678.
- [74] Pierre Comon. “Independent component analysis, a new concept?” In: *Signal processing* 36.3 (1994), pp. 287–314.
- [75] Aapo Hyvärinen. “Independent component analysis: recent advances”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 371.1984 (2013), p. 20110534.
- [76] Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780.
- [77] Dongda Zhang, Thomas R Savage, and Bovinille A Cho. “Combining model structure identification and hybrid modelling for photo-production process predictive simulation and optimisation”. In: *Biotechnology and Bioengineering* 117.11 (2020), pp. 3356–3367.
- [78] Rongxuan Wang et al. “Real-time process monitoring and closed-loop control on laser power via a customized laser powder bed fusion platform”. In: *Additive Manufacturing* 66 (2023), p. 103449.
- [79] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. “A training algorithm for optimal margin classifiers”. In: *Proceedings of the fifth annual workshop on Computational learning theory*. 1992, pp. 144–152.
- [80] Shen Yin et al. “A review on basic data-driven approaches for industrial process monitoring”. In: *IEEE Transactions on Industrial electronics* 61.11 (2014), pp. 6418–6428.
- [81] Mehdi Namdari and Hooshang Jazayeri-Rad. “Incipient fault diagnosis using support vector machines based on monitoring continuous decision functions”. In: *Engineering Applications of Artificial Intelligence* 28 (2014), pp. 22–35.
- [82] Davi L de Souza et al. “Fault detection and diagnosis using support vector machines-a SVC and SVR comparison”. In: *J. Saf. Eng* 3.1 (2014), pp. 18–29.
- [83] Roel Bouman, Zaharah Bukhsh, and Tom Heskes. “Unsupervised anomaly detection algorithms on real-world data: How many do we need?” In: *Journal of Machine Learning Research* 25 (2024), pp. 1–55.

- [84] Markus M Breunig et al. “LOF: identifying density-based local outliers”. In: *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. 2000, pp. 93–104.
- [85] Martin Ester et al. “A density-based algorithm for discovering clusters in large spatial databases with noise”. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*. AAAI Press, 1996, pp. 226–231.
- [86] Yuhang Gong et al. “RT-DBSCAN: Real-time parallel clustering of spatio-temporal data using spark streaming”. In: *International Conference on Computational Science*. Springer. 2018, pp. 523–536.
- [87] Seema Bandyopadhyay and Edward J Coyle. “An energy efficient hierarchical clustering algorithm for wireless sensor networks”. In: *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies*. Vol. 3. IEEE. 2003, pp. 1713–1723.
- [88] Yajie Cui, Zhaoxiang Liu, and Shiguo Lian. “A survey on unsupervised anomaly detection algorithms for industrial images”. In: *IEEE Access* 11 (2023), pp. 55297–55315. DOI: 10.1109/ACCESS.2023.3282993.
- [89] Chen Zhang et al. “Deep reinforcement learning for intelligent transportation systems: A survey”. In: *IEEE Transactions on Intelligent Transportation Systems* 21.11 (2019), pp. 4649–4666.
- [90] Yuxuan Liu, Huan He, and Zhi Li. “A reinforcement learning-based approach for intelligent resource allocation in edge computing”. In: *IEEE Transactions on Network and Service Management* 17.4 (2020), pp. 2068–2081.
- [91] Yuxiang Si et al. “A review of data-driven approaches for prediction and classification of bearing faults using machine learning and deep learning”. In: *Mechanical Systems and Signal Processing* 155 (2022), p. 107624.
- [92] Steven CH Hoi et al. “Online learning: A comprehensive survey”. In: *Neurocomputing* 459 (2021), pp. 249–289.
- [93] Maria Weese et al. “Statistical learning methods applied to process monitoring: An overview and perspective”. In: *Journal of Quality Technology* 48.1 (2016), pp. 4–24.
- [94] Humayra Islam and Tarem Ahmed. “Anomaly clustering based on correspondence analysis”. In: *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*. IEEE. 2018, pp. 1019–1025.
- [95] Xiaojin Zhu and Zoubin Ghahramani. “Learning from labeled and unlabeled data with label propagation”. In: *ProQuest Number: INFORMATION TO ALL USERS* (2002).
- [96] Charu C Aggarwal. “Mining sensor data streams”. In: *Managing and mining sensor data* (2013), pp. 143–171.
- [97] Yanhao Wang et al. “Real-time influence maximization on dynamic social streams”. In: *arXiv preprint arXiv:1702.01586* (2017).
- [98] Vinicius MA Souza et al. “Challenges in benchmarking stream learning algorithms with real-world data”. In: *Data Mining and Knowledge Discovery* 34.6 (2020), pp. 1805–1858.

- [99] Andrew McAfee et al. "Big data: the management revolution". In: *Harvard business review* 90.10 (2012), pp. 60–68.
- [100] Min Chen, Shiwen Mao, and Yunhao Liu. "Big data: A survey". In: *Mobile networks and applications* 19 (2014), pp. 171–209.
- [101] Benjamin T Hazen et al. "Data quality for data science, predictive analytics, and big data in supply chain management: An introduction to the problem and suggestions for research and applications". In: *International Journal of Production Economics* 154 (2014), pp. 72–80.
- [102] Muhammad Naeem et al. "Trends and future perspective challenges in big data". In: *Advances in Intelligent Data Analysis and Applications: Proceeding of the Sixth Euro-China Conference on Intelligent Data Analysis and Applications, 15–18 October 2019, Arad, Romania*. Springer. 2022, pp. 309–325.
- [103] Chunquan Li, Yaqiong Chen, and Yuling Shang. "A review of industrial big data for decision making in intelligent manufacturing". In: *Engineering Science and Technology, an International Journal* 29 (2022), p. 101021.
- [104] Yichuan Wang, LeeAnn Kung, and Terry Anthony Byrd. "Big data analytics: Understanding its capabilities and potential benefits for healthcare organizations". In: *Technological forecasting and social change* 126 (2018), pp. 3–13.
- [105] Halima Oluwabunmi Bello, Courage Idemudia, and Toluwalase Vanessa Iyelolu. "Integrating machine learning and blockchain: Conceptual frameworks for real-time fraud detection and prevention". In: *World Journal of Advanced Research and Reviews* 23.1 (2024), pp. 056–068.
- [106] Naser Kalid et al. "Based real time remote health monitoring systems: A review on patients prioritization and related "big data" using body sensors information and communication technology". In: *Journal of medical systems* 42 (2018), pp. 1–30.
- [107] Bibitayo Ebunlomo Abikoye et al. "Real-time financial monitoring systems: Enhancing risk management through continuous oversight". In: *GSC Advanced Research and Reviews* 20.1 (2024), pp. 465–476.
- [108] NG Yen Ting, TAN Yee Shee, and LOW Jonathan Sze Choong. "Internet of things for real-time waste monitoring and benchmarking: waste reduction in manufacturing shop floor". In: *Procedia CIRP* 61 (2017), pp. 382–386.
- [109] Muhammad Danish. "Enhancing Cyber Security through Predictive Analytics: Real-Time Threat Detection and Response". In: *arXiv preprint arXiv:2407.10864* (2024).
- [110] Galit B Yom-Tov et al. "Customer sentiment in web-based service interactions: Automated analyses and new insights". In: *Companion Proceedings of the The Web Conference 2018*. 2018, pp. 1689–1697.
- [111] Sina Djali et al. "How a data-driven quality management system can manage compliance risk in clinical trials". In: *Drug Information Journal* 44.4 (2010), pp. 359–373.
- [112] Jasmin Praful Bharadiya. "Machine learning and AI in business intelligence: Trends and opportunities". In: *International Journal of Computer (IJC)* 48.1 (2023), pp. 123–134.

- [113] David L Donoho et al. “High-dimensional data analysis: The curses and blessings of dimensionality”. In: *AMS math challenges lecture 1.2000* (2000), p. 32.
- [114] Zahra Zohrevand. “End-to-end anomaly detection in stream data”. In: *Simon Fraser Universit* (2020).
- [115] Naomi Altman and Martin Krzywinski. “The curse (s) of dimensionality”. In: *Nat Methods* 15.6 (2018), pp. 399–400.
- [116] Tathagata Das et al. “Adaptive stream processing using dynamic batch sizing”. In: *Proceedings of the ACM Symposium on Cloud Computing*. 2014, pp. 1–13.
- [117] Bin Zhou. “High-frequency data and volatility in foreign-exchange rates”. In: *Journal of Business & Economic Statistics* 14.1 (1996), pp. 45–52.
- [118] Christian Capezza et al. “Robust multivariate functional control chart”. In: *Technometrics* 66.4 (2024), pp. 531–547.
- [119] Aggeliki Sgora Sotiris Bersimis and Stelios Psarakis. “The application of multivariate statistical process monitoring in non-industrial processes”. In: *Quality Technology & Quantitative Management* 15.4 (2018), pp. 526–549. DOI: 10.1080/16843703.2016.1226711.
- [120] Mandla D Diko, Subha Chakraborti, and Marien Alet Graham. “Monitoring the process mean when standards are unknown: A classic problem revisited”. In: *Quality and Reliability Engineering International* 32.2 (2016), pp. 609–622.
- [121] Maroua Bahri et al. “Data stream analysis: Foundations, major tasks and tools”. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 11.3 (2021), e1405.
- [122] Riyaz Ahamed Ariyaluran Habeeb et al. “Real-time big data processing for anomaly detection: A survey”. In: *International Journal of Information Management* 45 (2019), pp. 289–307.
- [123] Sergio Ramírez-Gallego et al. “A survey on data preprocessing for data stream mining: Current status and future directions”. In: *Neurocomputing* 239 (2017), pp. 39–57.
- [124] Harold Hotelling et al. “The generalization of Student’s ratio”. In: *Breakthroughs in Statistics* (1931).
- [125] Hasan Bulut. “A robust Hotelling test statistic for one sample case in high dimensional data”. In: *Communications in Statistics-Theory and Methods* 52.13 (2023), pp. 4590–4604.
- [126] Alireza Faraz et al. “The robust economic statistical design of the hotelling’s T2 chart”. In: *Communications in statistics-Theory and methods* 45.23 (2016), pp. 6989–7001.
- [127] Rebecca Zwick. “Rank and normal scores alternatives to Hotelling’s T2”. In: *Multivariate Behavioral Research* 21.2 (1986), pp. 169–186.
- [128] Momiao Xiong, Jinying Zhao, and Eric Boerwinkle. “Generalized T2 test for genome association studies”. In: *The American Journal of Human Genetics* 70.5 (2002), pp. 1257–1268.
- [129] AD Pouliezios and George S Stavrakakis. *Real time fault monitoring of industrial processes*. Vol. 12. Springer Science & Business Media, 2013.

- [130] Jessamyn Liu. “Anomaly detection methods for detecting cyber attacks in industrial control systems”. PhD thesis. Massachusetts Institute of Technology, 2020.
- [131] Zongliang Hu, Tiejun Tong, and Marc G Genton. “A Pairwise Hotelling Method for Testing High-Dimensional Mean Vectors”. In: *arXiv preprint arXiv:2003.04636* (2020).
- [132] MJ Healy. “Statistics from the inside. 12. Non-normal data.” In: *Archives of disease in childhood* 70.2 (1994), p. 158.
- [133] Dai Deliang. “On high-dimensional Mahalanobis distances”. PhD thesis. Linnaeus University Press, 2017.
- [134] Xiufeng Liu, Nadeem Iftikhar, and Xike Xie. “Survey of real-time processing systems for big data”. In: *Proceedings of the 18th International Database Engineering & Applications Symposium*. 2014, pp. 356–361.
- [135] Dewi Nur Aisyah et al. “The use of a health compliance monitoring system during the COVID-19 pandemic in Indonesia: evaluation study”. In: *JMIR Public Health and Surveillance* 8.11 (2022), e40089.
- [136] Nikhil R Pal and J Biswas. “Cluster validation using graph theoretic concepts”. In: *Pattern Recognition* 30.6 (1997), pp. 847–857.
- [137] José María Luna-Romera et al. “An approach to validity indices for clustering techniques in big data”. In: *Progress in Artificial Intelligence* 7 (2018), pp. 81–94.
- [138] Richard O Duda, Peter E Hart, et al. *Pattern classification*. John Wiley & Sons, 2006.
- [139] Ranjan K Mallik. “On multivariate Rayleigh and exponential distributions”. In: *IEEE Transactions on Information Theory* 49.6 (2003), pp. 1499–1515.
- [140] Kazutomo Kawamura. “The structure of multivariate Poisson distribution”. In: *Kodai Mathematical Journal* 2.3 (1979), pp. 337–345.
- [141] Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. “On the surprising behavior of distance metrics in high dimensional space”. In: *Database theory—ICDT 2001: 8th international conference London, UK, January 4–6, 2001 proceedings* 8. Springer. 2001, pp. 420–434.
- [142] Masud Moshtaghi et al. “Online cluster validity indices for performance monitoring of streaming data clustering”. In: *International Journal of Intelligent Systems* 34.4 (2019), pp. 541–563.
- [143] Cynthia A Lowry and Douglas C Montgomery. “A review of multivariate control charts”. In: *IIE transactions* 27.6 (1995), pp. 800–810.
- [144] Nicoleta Tantalaki, Stavros Souravlas, and Manos Roumeliotis. “A review on big data real-time stream processing and its scheduling techniques”. In: *International Journal of Parallel, Emergent and Distributed Systems* 35.5 (2020), pp. 571–601.
- [145] Kristof Kovacs and Andrew RA Conway. “Process overlap theory: A unified account of the general factor of intelligence”. In: *Psychological Inquiry* 27.3 (2016), pp. 151–177.
- [146] Walter A Shewhart and W Edwards Deming. “In Memoriam: Walter A. Shewhart, 1891–1967”. In: *The American Statistician* 21.2 (1967), pp. 39–40.

- [147] Walter Andrew Shewhart and William Edwards Deming. *Statistical method from the viewpoint of quality control*. Courier Corporation, 1986.
- [148] John F MacGregor and Theodora Kourti. “Statistical process control of multivariate processes”. In: *Control engineering practice* 3.3 (1995), pp. 403–414.
- [149] Sotiris Bersimis, John Panaretos, and Stelios Psarakis. “Multivariate statistical process control charts and the problem of interpretation: a short overview and some applications in industry”. In: *arXiv preprint arXiv:0901.2880* (2009).
- [150] Tadeusz Caliński and Jerzy Harabasz. “A dendrite method for cluster analysis”. In: *Communications in Statistics-theory and Methods* 3.1 (1974), pp. 1–27.
- [151] David L Davies and Donald W Bouldin. “A cluster separation measure”. In: *IEEE transactions on pattern analysis and machine intelligence* 2 (2009), pp. 224–227.
- [152] Kristen B Gorman, Tony D Williams, and William R Fraser. “Ecological sexual dimorphism and environmental variability within a community of Antarctic penguins (genus *Pygoscelis*)”. In: *PloS one* 9.3 (2014), e90081.
- [153] Allison Marie Horst, Alison Presmanes Hill, and Kristen B Gorman. *palmer-penguins: Palmer Archipelago (Antarctica) penguin data*. R package version 0.1.0. 2020. DOI: 10.5281/zenodo.3960218. URL: <https://allisonhorst.github.io/palmerpenguins/>.
- [154] Oresti Banos et al. “mHealthDroid: a novel framework for agile development of mobile health applications”. In: *International workshop on ambient assisted living*. Springer. 2014, pp. 91–98.
- [155] Dominic Masters and Carlo Luschi. “Revisiting small batch training for deep neural networks”. In: *arXiv preprint arXiv:1804.07612* (2018).
- [156] Zhou Wu and Xiaohua Xia. “Tariff-driven demand side management of green ship”. In: *Solar Energy* 170 (2018), pp. 991–1000.
- [157] Jasper Faber et al. “Fourth IMO greenhouse gas study 2020”. In: *International Maritime Organization* (2020), pp. 1689–1699.
- [158] International Maritime Organization. *MEPC 75/18/Add.1. Resolution MEPC.324(75). Amendments to MARPOL Annex VI*. Tech. rep. London, United Kingdom: International Maritime Organization, 2020.
- [159] Zabi Bazari and Tore Longva. “Assessment of IMO mandated energy efficiency measures for international shipping”. In: *International Maritime Organization* 10 (2011).
- [160] Nishatabbas Rehmatulla, John Calleya, and Tristan Smith. “The implementation of technical energy efficiency and CO2 emission reduction measures in shipping”. In: *Ocean engineering* 139 (2017), pp. 184–197.
- [161] Martin Gibson, Alan J Murphy, and Kayvan Pazouki. “Evaluation of environmental performance indices for ships”. In: *Transportation Research Part D: Transport and Environment* 73 (2019), pp. 152–161.
- [162] Tobias Haas and Hendrik Sander. “Decarbonizing transport in the European Union: Emission performance standards and the perspectives for a European Green Deal”. In: *Sustainability* 12.20 (2020), p. 8381.

- [163] DG Trodden et al. “Fuel usage data analysis for efficient shipping operations”. In: *Ocean Engineering* 110 (2015), pp. 75–84.
- [164] Jane Lister, René Taudal Poulsen, and Stefano Ponte. “Orchestrating transnational environmental governance in maritime shipping”. In: *Global Environmental Change* 34 (2015), pp. 185–195.
- [165] EV Telfer. *The practical analysis of merchant ship trials and service performance*. North east coast institution of engineers and shipbuilders, 1927.
- [166] John F Messerly et al. “Low-temperature thermal data for pentane, n-heptadecane, and n-octadecane. Revised thermodynamic functions for the n-alkanes, C5-C18”. In: *Journal of Chemical and Engineering Data* 12.3 (1967), pp. 338–346.
- [167] Roar Adland et al. “The energy efficiency effects of periodic ship hull cleaning”. In: *Journal of Cleaner Production* 178 (2018), pp. 1–13.
- [168] Yuquan Du et al. “Two-phase optimal solutions for ship speed and trim optimization over a voyage using voyage report data”. In: *Transportation Research Part B: Methodological* 122 (2019), pp. 88–114.
- [169] Paul Balcombe et al. “How to decarbonise international shipping: Options for fuels, technologies and policies”. In: *Energy conversion and management* 182 (2019), pp. 72–88.
- [170] Suzanne Greene, Haiying Jia, and Gabriela Rubio-Domingo. “Well-to-tank carbon emissions from crude oil maritime transportation”. In: *Transportation Research Part D: Transport and Environment* 88 (2020), p. 102587.
- [171] Michael R Motley, Mayer Nelson, and Yin L Young. “Integrated probabilistic design of marine propulsors to minimize lifetime fuel consumption”. In: *Ocean Engineering* 45 (2012), pp. 1–8.
- [172] G Doulgeris et al. “Techno-economic and environmental risk analysis for advanced marine propulsion systems”. In: *Applied energy* 99 (2012), pp. 1–12.
- [173] AJ Murphy et al. “Modelling ship emission factors and emission indices”. In: *Low carbon shipping conference*. Newcastle University. 2013.
- [174] Joanne Scott et al. “The promise and limits of private standards in reducing greenhouse gas emissions from shipping”. In: *Journal of Environmental Law* 29.2 (2017), pp. 231–262.
- [175] TWP Smith, S Parker, and N Rehmatulla. “On the speed of ships”. In: *International Conference on Technologies, Operations, Logistics and Modelling for Low Carbon Shipping, LCS2011*. University of Strathclyde, Glasgow, UK. 2011, pp. 22–24.
- [176] C Capezza et al. “Ship fuel consumption monitoring and fault detection via partial least squares and control charts of navigation data”. In: *Transportation Research Part D: Transport and Environment* 67 (2019), pp. 375–387.
- [177] Fabian Tillig et al. “Analysis of uncertainties in the prediction of ships’ fuel consumption—from early design to operation conditions”. In: *Ships and Offshore Structures* 13.sup1 (2018), pp. 13–24.
- [178] Michael I Jordan and Tom M Mitchell. “Machine learning: Trends, perspectives, and prospects”. In: *Science* 349.6245 (2015), pp. 255–260.

- [179] AI Parkes, AJ Sobey, and DA Hudson. “Physics-based shaft power prediction for large merchant ships using neural networks”. In: *Ocean Engineering* 166 (2018), pp. 92–104.
- [180] Chi Zhang et al. “Data-driven ship energy efficiency analysis and optimization model for route planning in ice-covered Arctic waters”. In: *Ocean Engineering* 186 (2019), p. 106071.
- [181] Lúcia Moreira, Roberto Vettor, and Carlos Guedes Soares. “Neural network approach for predicting ship speed and fuel consumption”. In: *Journal of Marine Science and Engineering* 9.2 (2021), p. 119.
- [182] Pavlos Karagiannidis and Nikos Themelis. “Data-driven modelling of ship propulsion and the effect of data pre-processing on the prediction of ship fuel consumption and speed loss”. In: *Ocean Engineering* 222 (2021), p. 108616.
- [183] Onder Canbulat et al. “An application of BBNs on the integrated energy efficiency of ship–port interface: a dry bulk shipping case”. In: *Maritime Policy & Management* 46.7 (2019), pp. 845–865.
- [184] Young-Rong Kim, Min Jung, and Jun-Bum Park. “Development of a fuel consumption prediction model based on machine learning using ship in-service data”. In: *Journal of Marine Science and Engineering* 9.2 (2021), p. 137.
- [185] Zhihui Hu et al. “Prediction of fuel consumption for enroute ship based on machine learning”. In: *IEEE Access* 7 (2019), pp. 119497–119505.
- [186] Xiao Lang, Da Wu, and Wengang Mao. “Comparison of supervised machine learning methods to predict ship propulsion power at sea”. In: *Ocean Engineering* 245 (2022), p. 110387.
- [187] Tristan Smith et al. *Third IMO Greenhouse Gas Study 2014*. London, United Kingdom: International Maritime Organization, 2015.
- [188] Bedřich Moldan, Svatava Janoušková, and Tomáš Hák. “How to understand and measure environmental sustainability: Indicators and targets”. In: *Ecological indicators* 17 (2012), pp. 4–13.
- [189] S Afshin Mansouri, Habin Lee, and Oluwakayode Aluko. “Multi-objective decision support to enhance environmental sustainability in maritime shipping: A review and future directions”. In: *Transportation Research Part E: Logistics and Transportation Review* 78 (2015), pp. 3–18.
- [190] Sotiria Lagouvardou, Harilaos N Psaraftis, and Thalys Zis. “A literature survey on market-based measures for the decarbonization of shipping”. In: *Sustainability* 12.10 (2020), p. 3953.
- [191] Harilaos N Psaraftis and Christos A Kontovas. “Speed models for energy-efficient maritime transportation: A taxonomy and survey”. In: *Transportation Research Part C: Emerging Technologies* 26 (2013), pp. 331–351.
- [192] Jun Xia et al. “Joint planning of fleet deployment, speed optimization, and cargo allocation for liner shipping”. In: *Transportation Science* 49.4 (2015), pp. 922–938.
- [193] Qie He, Xiaochen Zhang, and Kameng Nip. “Speed optimization over a path with heterogeneous arc costs”. In: *Transportation Research Part B: Methodological* 104 (2017), pp. 198–214.

- [194] Benjamin Pjedsted Pedersen and Jan Larsen. “Prediction of full-scale propulsion power using artificial neural networks”. In: *Proceedings of the 8th international conference on computer and IT applications in the maritime industries (COM-PIT’09), Budapest, Hungary May. 2009*, pp. 10–12.
- [195] Roar Adland, Pierre Cariou, and Francois-Charles Wolff. “Optimal ship speed and the cubic law revisited: Empirical evidence from an oil tanker fleet”. In: *Transportation Research Part E: Logistics and Transportation Review* 140 (2020), p. 101972.
- [196] Dong Yang et al. “How big data enriches maritime research—a critical review of Automatic Identification System (AIS) data applications”. In: *Transport Reviews* 39.6 (2019), pp. 755–773.
- [197] E Bal Beşikçi et al. “An artificial neural network based decision support system for energy efficient ship operations”. In: *Computers & Operations Research* 66 (2016), pp. 393–401.
- [198] Miyeon Jeon et al. “Prediction of ship fuel consumption by using an artificial neural network”. In: *Journal of Mechanical Science and Technology* 32 (2018), pp. 5785–5796.
- [199] Despoina Dede et al. “A Statistical Framework for Assessing Environmental Performance of Quality Wine Production”. In: *Sustainability* 12.24 (2020), p. 10246.
- [200] Sotirios Bersimis and Dimitrios Georgakellos. “A probabilistic framework for the evaluation of products’ environmental performance using life cycle approach and Principal Component Analysis”. In: *Journal of Cleaner Production* 42 (2013), pp. 103–115.
- [201] JE Jackson and A Edward. “User’s guide to principal components. John Wiley Sons”. In: *Inc., New York* 40 (1991).
- [202] William Feller. *An introduction to probability theory and its applications*. Tech. rep. Wiley series in probability and mathematical statistics, 3rd edn.(Wiley, New York, 1967.
- [203] Robert Tibshirani, Guenther Walther, and Trevor Hastie. “Estimating the number of clusters in a data set via the gap statistic”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 63.2 (2001), pp. 411–423.
- [204] Amira M El-Mandouh et al. “Optimized K-means clustering model based on gap statistic”. In: *International Journal of Advanced Computer Science and Applications* 10.1 (2019).
- [205] Andrew Rawson et al. “A machine learning approach for monitoring ship safety in extreme weather events”. In: *Safety science* 141 (2021), p. 105336.
- [206] Emre Akyuz, Kadir Cicek, and Metin Celik. “A comparative research of machine learning impact to future of maritime transportation”. In: *Procedia Computer Science* 158 (2019), p. 275.
- [207] Funda Kaya İnceişçi and AK Ayça. “Fault Analysis of Ship Machinery Using Machine Learning Techniques”. In: *International Journal of Maritime Engineering* 164.A1 (2022).

- [208] Taewoong Hwang and Ik-Hyun Youn. “Navigation Situation Clustering Model of Human-Operated Ships for Maritime Autonomous Surface Ship Collision Avoidance Tests”. In: *Journal of Marine Science and Engineering* 9.12 (2021), p. 1458.
- [209] Shamsudeen Temitope Yekeen, Abdul-Lateef Balogun, and Khamaruzaman B Wan Yusof. “A novel deep learning instance segmentation model for automated marine oil spill detection”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 167 (2020), pp. 190–200.
- [210] Tayfun Uyanik, Çağlar Karatuğ, and Yasin Arslanoğlu. “Machine learning approach to ship fuel consumption: A case of container vessel”. In: *Transportation Research Part D: Transport and Environment* 84 (2020), p. 102389.
- [211] Luofeng Huang et al. “Machine learning in sustainable ship design and operation: A review”. In: *Ocean Engineering* 266 (2022), p. 112907.
- [212] Yuquan Du et al. “Data fusion and machine learning for ship fuel efficiency modeling: Part III–Sensor data and meteorological data”. In: *Communications in Transportation Research* 2 (2022), p. 100072.
- [213] Aybars Oruc. “Claims of state-sponsored cyberattack in the maritime industry”. In: *Conference Proceedings of INEC*. 2020.
- [214] Choong Bae Lee et al. “A cross-country study of competitiveness of the shipping industry”. In: *Transport Policy* 35 (2014), pp. 366–376.
- [215] Ibna Zaman et al. “Challenges and opportunities of big data analytics for upcoming regulations and future transformation of the shipping industry”. In: *Procedia engineering* 194 (2017), pp. 537–544.
- [216] Khanh Q Bui and Lokukaluge P Perera. “The compliance challenges in emissions control regulations to reduce air pollution from shipping”. In: *OCEANS 2019-Marseille*. IEEE. 2019, pp. 1–8.
- [217] Albert Buixadé Farré et al. “Commercial Arctic shipping through the Northeast Passage: routes, resources, governance, technology, and infrastructure”. In: *Polar Geography* 37.4 (2014), pp. 298–324.
- [218] Mohammad Danil Arifin. “Application of Internet of Things (IoT) and Big Data in the Maritime Industries: Ship Allocation Model”. In: *International Journal of Marine Engineering Innovation and Research* 8.1 (2023).
- [219] Iain Shepherd. “European efforts to make marine data more accessible”. In: *Ethics in Science and Environmental Politics* 18 (2018), pp. 75–81.
- [220] Andrew Rawson and Mario Brito. “A survey of the opportunities and challenges of supervised machine learning in maritime risk analysis”. In: *Transport Reviews* 43.1 (2023), p. 108.
- [221] Eleftheria Eliopoulou, Apostolos Papanikolaou, and Markos Voulgarellis. “Statistical analysis of ship accidents and review of safety level”. In: *Safety science* 85 (2016), pp. 282–292.
- [222] Julio Vergara, Chris McKesson, and Magdalena Walczak. “Sustainable energy for the marine sector”. In: *Energy Policy* 49 (2012), pp. 333–345.

- [223] Hakan Akyildiz and Ayhan Menten. “An integrated risk assessment based on uncertainty analysis for cargo vessel safety”. In: *Safety science* 92 (2017), pp. 34–43.
- [224] Christiaan Heij and Sabine Knapp. “Evaluation of safety and environmental risk at individual ship and company level”. In: *Transportation research part D: transport and environment* 17.3 (2012), pp. 228–236.
- [225] Jihong Chen et al. “Identifying factors influencing total-loss marine accidents in the world: Analysis and evaluation based on ship types and sea regions”. In: *Ocean Engineering* 191 (2019), p. 106495.
- [226] Chao-Wei Lu et al. “Marine Collision Avoidance Route Planning Model for MASS Based on Domain-Based Predicted Area of Danger”. In: *Journal of Marine Science and Engineering* 11.9 (2023), p. 1724.
- [227] Marie Rydenfelt. “Insurances in Ship Finance”. In: (2013), pp. 73–74.
- [228] Kipkemoi J Ngeno and Hiroshi Watanabe. “Ship Identification for Port Security using Faster Region Convolution Neural Network”. In: *IEICE General Conference, BS-1*. Vol. 12. 2017.
- [229] Tlameo Emmanuel et al. “A survey on missing data in machine learning”. In: *Journal of Big Data* 8.1 (2021), pp. 1–37.
- [230] Donia Smaali Bouhlila and Fethi Sellaouti. “Multiple imputation using chained equations for missing data in TIMSS: a case study”. In: *Large-scale Assessments in Education* 1 (2013), pp. 1–33.
- [231] Kimseth Seu, Mi-Sun Kang, and HwaMin Lee. “An intelligent missing data imputation techniques: A review”. In: *JOIV: International Journal on Informatics Visualization* 6.1-2 (2022), pp. 278–283.
- [232] Antonia J. Henry et al. “Comparative methods for handling missing data in large databases”. In: *Journal of Vascular Surgery* 58.5 (2013), 1353–1359.e6. ISSN: 0741-5214.
- [233] Roderick JA Little. “A test of missing completely at random for multivariate data with missing values”. In: *Journal of the American statistical Association* 83.404 (1988), pp. 1198–1202.
- [234] Ali Shehadeh et al. “Machine learning models for predicting the residual value of heavy construction equipment: An evaluation of modified decision tree, LightGBM, and XGBoost regression”. In: *Automation in Construction* 129 (2021), p. 103827.
- [235] Selvaprabu Jeganathan et al. “Predicting Academic Performance of Immigrant Students Using XGBoost Regressor”. In: *International Journal of Information Technology and Web Engineering (IJITWE)* 17.1 (2022), pp. 1–19.
- [236] Mjimer Imane, Es-Saadia Aoula, and El Hassan Achouyab. “Using Bayesian ridge regression to predict the overall equipment effectiveness performance”. In: *2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*. IEEE. 2022, pp. 1–4.
- [237] Alexei Botchkarev. “Evaluating performance of regression machine learning models using multiple error metrics in azure machine learning studio”. In: *Available at SSRN 3177507* (2018).

- [238] Guy S Handelman et al. “Peering into the black box of artificial intelligence: evaluation metrics of machine learning methods”. In: *American Journal of Roentgenology* 212.1 (2019), pp. 38–43.
- [239] ES Bekri, PC Yannopoulos, and P Economou. “Methodology for improving reliability of river discharge measurements”. In: *Journal of environmental management* 247 (2019), pp. 371–384.
- [240] Panagiotis Alexopoulos et al. “Validation of the cognitive telephone screening instruments COGTEL and COGTEL+ in identifying clinically diagnosed neurocognitive disorder due to Alzheimer’s disease in a naturalistic clinical setting”. In: *Journal of Alzheimer’s Disease* 83.1 (2021), pp. 259–268.
- [241] Panagiotis K Tsikas, Athanasios P Chassiakos, and Vasileios C Papadimitropoulos. “Seismic damage assessment of highway bridges by means of soft computing techniques”. In: *Structure and Infrastructure Engineering* (2022), pp. 1–13.
- [242] Liping Zhang et al. “Spatiotemporal Evolution Characteristics of Urban Land Surface Temperature Based on Local Climate Zones in Xi’an Metropolitan, China”. In: *Chinese Geographical Science* (2023), pp. 1–16.
- [243] Polychronis Economou, Apostolos Batsidis, and Konstantinos Kounetas. “Evaluation of the OECD’s prediction algorithm for the annual GDP growth rate”. In: *Communications in Statistics: Case Studies, Data Analysis and Applications* 7.1 (2021), pp. 67–87.
- [244] S. Velliangiri, S. Alagumuthukrishnan, and S Iwin Thankumar joseph. “A Review of Dimensionality Reduction Techniques for Efficient Computation”. In: *Procedia Computer Science* 165 (2019), pp. 104–111.
- [245] J Edward Jackson. *A user’s guide to principal components*. John Wiley & Sons, 2005.
- [246] Sotiris Bersimis, Aggeliki Sgora, and Stelios Psarakis. “Methods for interpreting the out-of-control signal of multivariate control charts: A comparison study”. In: *Quality and Reliability Engineering International* 33.8 (2017), pp. 2295–2326.
- [247] PE Maravelakis et al. “Identifying the out of control variable in a multivariate control chart”. In: *Communications in Statistics-Theory and methods* 31.12 (2002), pp. 2391–2408.
- [248] Henry F Kaiser. “The application of electronic computers to factor analysis”. In: *Educational and psychological measurement* 20.1 (1960), pp. 141–151.
- [249] Leland McInnes, John Healy, and James Melville. “Umap: Uniform manifold approximation and projection for dimension reduction”. In: *arXiv preprint arXiv:1802.03426* (2018).
- [250] Marc Vermeulen et al. “Application of Uniform Manifold Approximation and Projection (UMAP) in spectral imaging of artworks”. In: *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy* 252 (2021), p. 119547.
- [251] Djuradj Milošević et al. “The application of Uniform Manifold Approximation and Projection (UMAP) for unconstrained ordination and classification of biological indicators in aquatic ecology”. In: *Science of The Total Environment* 815 (2022), p. 152365.

- [252] Teng-To Yu et al. “Application of high-dimensional uniform manifold approximation and projection (UMAP) to cluster existing landfills on the basis of geographical and environmental features”. In: *Science of The Total Environment* 904 (2023), p. 167013.
- [253] Petros E Maravelakis and Sotirios Bersimis. “The use of Andrews curves for detecting the out-of-control variables when a multivariate control chart signals”. In: *Statistical Papers* 50 (2009), pp. 51–65.
- [254] E Skamnia et al. “Hot spot identification method based on Andrews curves: an application on the COVID-19 crisis effects on caregiver distress in neurocognitive disorder”. In: *Journal of Applied Statistics* 50.11-12 (2023), pp. 2388–2407.
- [255] Philippe Hamel and Douglas Eck. “Learning features from music audio with deep belief networks.” In: *ISMIR*. Vol. 10. Utrecht, The Netherlands. 2010, p. 341.
- [256] Mehala Balamurali, Katherine L Silversides, and Arman Melkumyan. “A comparison of t-SNE, SOM and SPADE for identifying material type domains in geological data”. In: *Computers & Geosciences* 125 (2019), pp. 78–89.
- [257] Mehala Balamurali and Arman Melkumyan. “t-SNE based visualisation and clustering of geological domain”. In: *Neural Information Processing: 23rd International Conference, ICONIP 2016, Kyoto, Japan, October 16–21, 2016, Proceedings, Part IV 23*. Springer. 2016, pp. 565–572.
- [258] Raymond Leung, Mehala Balamurali, and Arman Melkumyan. “Sample truncation strategies for outlier removal in geochemical data: the MCD robust distance approach versus t-SNE ensemble clustering”. In: *Mathematical Geosciences* 53 (2021), pp. 105–130.
- [259] Andrew R Jamieson et al. “Exploring nonlinear feature space dimension reduction and data representation in breast CADx with Laplacian eigenmaps and-SNE”. In: *Medical physics* 37.1 (2010), pp. 339–351.
- [260] Izhar Wallach and Ryan Lilien. “The protein–small-molecule database, a non-redundant structural resource for the analysis of protein-ligand binding”. In: *Bioinformatics* 25.5 (2009), pp. 615–620.
- [261] Javad Birjandtalab, M Baran Pouyan, and Mehrdad Nourani. “Nonlinear dimension reduction for EEG-based epileptic seizure detection”. In: *2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*. IEEE. 2016, pp. 595–598.
- [262] Geoffrey E Hinton and Sam Roweis. “Stochastic neighbor embedding”. In: *Advances in neural information processing systems* 15 (2002).
- [263] Laurens Van der Maaten and Geoffrey Hinton. “Visualizing data using t-SNE.” In: *Journal of machine learning research* 9.11 (2008).
- [264] Leo Breiman. “Random forests”. In: *Machine learning* 45 (2001), pp. 5–32.
- [265] Leo Breiman. *Arcing classifiers*. Tech. rep. Technical report, University of California, Department of Statistics, 1996.
- [266] Pierre Geurts, Damien Ernst, and Louis Wehenkel. “Extremely randomized trees”. In: *Machine learning* 63 (2006), pp. 3–42.

- [267] Thomas Bayes. “Naive bayes classifier”. In: *Article Sources and Contributors* (1968), pp. 1–9.
- [268] Oliver Kramer and Oliver Kramer. “K-nearest neighbors”. In: *Dimensionality reduction with unsupervised nearest neighbors* (2013), pp. 13–23.
- [269] Jianping Gou et al. “A local mean-based k-nearest centroid neighbor classifier”. In: *The Computer Journal* 55.9 (2012), pp. 1058–1071.
- [270] Guo-Xun Yuan, Chia-Hua Ho, and Chih-Jen Lin. “Recent advances of large-scale linear classification”. In: *Proceedings of the IEEE* 100.9 (2012), pp. 2584–2603.
- [271] Koby Crammer et al. “Online passive aggressive algorithms”. In: *Journal of Machine Learning Research* 7 (2006), pp. 551–585.
- [272] Leo Breiman. “Pasting small votes for classification in large databases and on-line”. In: *Machine learning* 36 (1999), pp. 85–103.
- [273] Leo Breiman. “Bagging predictors”. In: *Machine learning* 24 (1996), pp. 123–140.
- [274] Tin Kam Ho. “The random subspace method for constructing decision forests”. In: *IEEE transactions on pattern analysis and machine intelligence* 20.8 (1998), pp. 832–844.
- [275] Gilles Louppe and Pierre Geurts. “Ensembles on random patches”. In: *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2012, Bristol, UK, September 24–28, 2012. Proceedings, Part I* 23. Springer. 2012, pp. 346–361.
- [276] Artur J Ferreira and Mário AT Figueiredo. “Boosting algorithms: A review of methods, theory, and applications”. In: *Ensemble machine learning: Methods and applications* (2012), pp. 35–85.
- [277] Ivan Jordanov, Nedyalko Petrov, and Alessio Petrozziello. “Classifiers Accuracy Improvement Based on Missing Data Imputation”. In: *Journal of Artificial Intelligence and Soft Computing Research* 8.1 (2018), pp. 31–48.
- [278] Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data*. Vol. 793. John Wiley & Sons, 2019.
- [279] Marco Ramoni and Paola Sebastiani. “Robust Bayes classifiers”. In: *Artificial Intelligence* 125.1 (2001), pp. 209–226.
- [280] Xunan Zhang, Shiji Song, and Cheng Wu. “Robust bayesian classification with incomplete data”. In: *Cognitive Computation* 5 (2013), pp. 170–187.
- [281] Isabelle Guyon et al. “Practical feature selection: from correlation to causality”. In: *Mining massive data sets for security: advances in data mining, search, social networks and text mining, and their applications to security* (2008), pp. 27–43.
- [282] Maira Anis and Mohsin Ali. “Investigating the performance of smote for class imbalanced learning: a case study of credit scoring datasets”. In: *Eur. Sci. J* 13.33 (2017), pp. 340–353.
- [283] Yotam Elor and Hadar Averbuch-Elor. “To SMOTE, or not to SMOTE?” In: *arXiv preprint arXiv:2201.08528* (2022).

- [284] Eslam A. Hussein et al. “Rainfall Prediction Using Machine Learning Models: Literature Survey”. In: *Artificial Intelligence for Data Science in Theory and Practice*. Ed. by Mohamed Alloghani, Christopher Thron, and Saad Subair. Cham: Springer International Publishing, 2022, pp. 75–108. ISBN: 978-3-030-92245-0. DOI: 10.1007/978-3-030-92245-0_4. URL: https://doi.org/10.1007/978-3-030-92245-0_4.
- [285] Sotiris Bersimis and Kostas Triantafyllopoulos. “Dynamic Non-parametric Monitoring of Air-Pollution”. In: *Methodology and Computing in Applied Probability* 22.4 (2020), pp. 1457–1479.
- [286] Rob J Hyndman and Yeasmin Khandakar. “Automatic time series forecasting: the forecast package for R”. In: *Journal of statistical software* 27 (2008), pp. 1–22.
- [287] Tae-Woong Yoo and Il-Seok Oh. “Time Series Forecasting of Agricultural Products’ Sales Volumes Based on Seasonal Long Short-Term Memory”. In: *Applied Sciences* 10.22 (2020).
- [288] A. Skamnia, E.S. Bekri, and P. Economou. “Main Diesel Engine Selection for Ships Based on Life Cycle Costing”. In: *Protection and Restoration of the Environment XVI. Kalamata*. Manariotis Ioannis et al. 2022, pp. 361–366.
- [289] Pouya Aghelpour, Vijay P Singh, and Vahid Varshavian. “Time series prediction of seasonal precipitation in Iran, using data-driven models: a comparison under different climatic conditions”. In: *Arabian Journal of Geosciences* 14.7 (2021), pp. 1–14.
- [290] Dongwoo Jang. “An Application of ANN Ensemble for Estimating of Precipitation Using Regional Climate Models”. In: *Advances in Civil Engineering* 2021 (2021).

