UNIVERISTY OF PIRAEUS - DEPARTMENT OF INFORMATICS

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ – ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

MSc «Advanced Informatics and Computing Systems - Software Development and Artificial Intelligence»

ΠΜΣ «Προηγμένα Συστήματα Πληροφορικής – Ανάπτυξη Λογισμικού και Τεχνητής Νοημοσύνης»

MSc Thesis

Μεταπτυχιακή Διατριβή

Thesis Title: Τίτλος Διατριβής:	**Ανάλυση και πρόβλεψη της ατμοσφαιρικής μόλυνσης σε πολυάριθμες χώρες μέσω χρήσης BiLSTM-Conv1D νευρωνικών δικτύων** Analysis and Prediction of air pollution in multiple countries using BiLSTM-Conv1D neural networks
Student’s name-surname: Ονοματεπώνυμο φοιτητή:	**GEORGIOS KARAMPELAS** ΓΕΩΡΓΙΟΣ ΚΑΡΑΜΠΕΛΑΣ
Father’s name: Πατρώνυμο:	**ILIAS** ΗΛΙΑΣ
Student’s ID No: Αριθμός Μητρώου:	**ΜΠΣΠ/19017**
Supervisor: Επιβλέπων:	**Dionisios Sotiropoulos, Assistant Professor** Διονύσιος Σωτηρόπουλος, Επίκουρος Καθηγητής

Νοέμβριος 2021/ November 2021

3-Member Examination Committee

Τριμελής Εξεταστική Επιτροπή

Dionisios Sotiropoulos **Assistant Professor**	**Georgios Tsihrintzis** **Professor**	**Evangelos Sakkopoulos** **Assistant Professor**
Διονύσιος Σωτηρόπουλος Επίκουρος Καθηγητής	Γεώργιος Τσιχριντζής Καθηγητής	Ευάγγελος Σακκόπουλος Επίκουρος Καθηγητής

Dedicated to my sister.

Abstract

Artificial Neural Networks is a scientific area, that has developed incrementally over the years and has been adopted to numerous fields of study due to its ability to process and discover patterns from raw data. The goal of a neural network model is to solve computational problems using different techniques and mathematical processes. One such problem is air pollution, an issue that is becoming more and more severe for the health of the general public. Governments and individuals require a way to know ahead of time and have an insight into how the quality of the air will be. That helps them take the initiative and act accordingly. The goal of this thesis is to develop a neural network model that will take as inputs historical data of the atmosphere, and it will be able to predict the future values of the pollutants that affect the quality of air. The implementation will be accomplished with the programming language Python and the usage of libraries that were developed for machine learning.

In the context of the master thesis, 6 research papers were studied for the different types of neural network models on data related to air pollution and time series forecasting. Furthermore, multiple different types of neural networks were developed in order to realize the final architecture of the proposed model. In addition, plentiful experimentation was conducted to determine the best hyperparameters for the model and was assessed both for its performance and accuracy but also for its generalization to more than one location around the globe.

Acknowledgements

The completion of this thesis would have been impossible without the guidance and support of my supervisor, Prof. Dionisios Sotiropoulos, whose valuable advice helped me to get out of the mud and find concrete results for the matter. From his consistent support and willingness to help me with such a deep and complex subject to undergo through, he has my utmost gratitude.

List of figures

List of tables

Contents

1. Introduction

1.1 Air Pollution

Air pollution refers to the discharge of pollutants into the air, which is detrimental to human health and to the planet. There are many types of air pollutants, both gases, and solids including ammonia (NH_3), carbon monoxide (CO), sulfur dioxide (SO_2), nitrous oxides (NOx), methane (CH_4), carbon dioxide (CO_2), chlorofluorocarbons (CFCs), particulates organic and inorganic, and biological molecules. Air pollution can cause allergies, diseases, and even death to both humans and other living organisms such as animals and food crops, and can damage the natural environment (climate change, ozone depletion, habitat degradation) or built environment (acid rain). human activity and natural processes can be responsible for generating air pollution.

According to the World Health Organization (WHO), every year air pollution is responsible for around seven million deaths worldwide. Currently, 9 out of 10 human beings breathe air that exceeds the WHO's guideline limits for pollutants, with those living in low- and middle-income countries suffering the most [1].

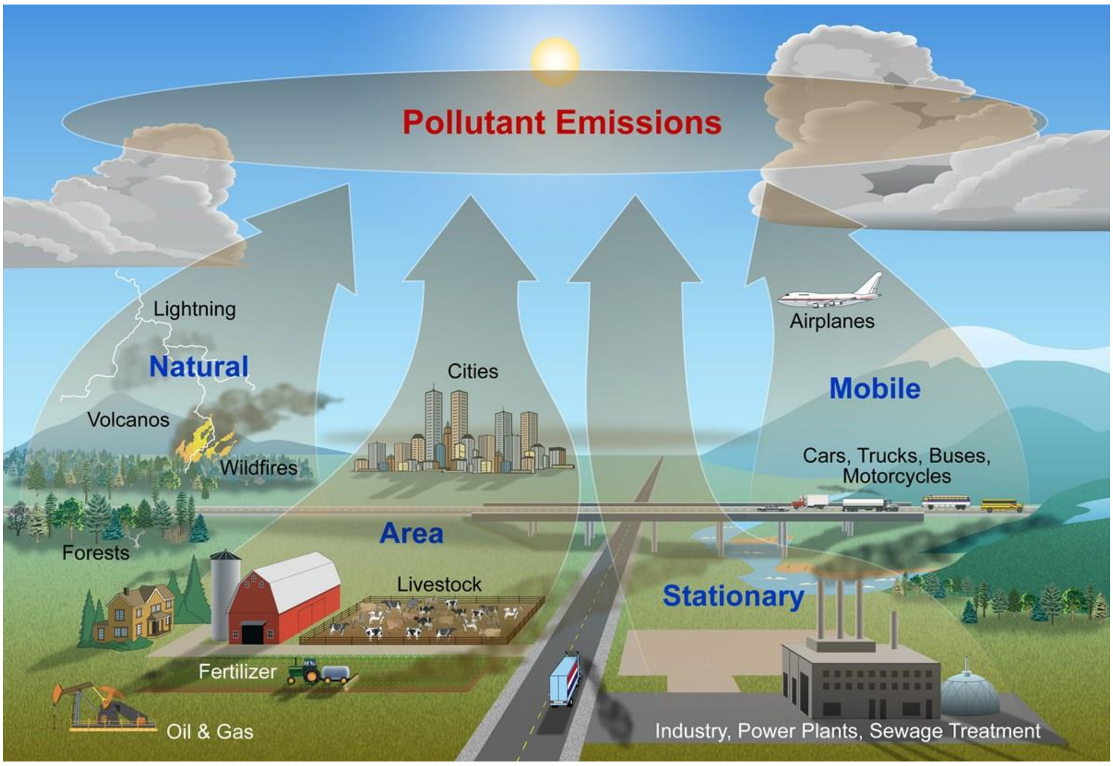


Figure 1. Multiple ways that pollutants go to the atmosphere [2].

The two most ubiquitous types of air pollution are Smog (ground-level ozone) which occurs when combusting fossil fuels emissions react with sunlight and Soot (particulate matter) which is made up of tiny particles of chemicals like soil, smoke, dust, or allergens that are carried in the air.

Smog can aggravate the eyes, throat, and damage the lungs, especially the youth, seniors, and people who work or exercise outdoors. It's even worse for people who have allergies, asthma, or other respiratory problems due to the fact that these extra pollutants can intensify their symptoms and trigger negative side effects.

Even the smallest airborne particles, either gaseous or solid in soot, are especially dangerous because they can penetrate the lungs and bloodstream and cause bronchitis, heart attacks, and even hasten death.

Several air pollutants pose severe health risks and can sometimes be fatal even in small amounts. Almost 200 of them are regulated by the law, with the most common being mercury, lead, dioxins, and benzene. Benzene can cause eye, skin, and lung irritation and even worse blood disorders in the long term. Dioxins, found usually in food but also within the air in small amounts, can affect the liver and in great exposure harm the immune, nervous, and endocrine systems as well as the reproductive functions. Mercury attacks the central nervous system. Large amounts of lead can damage a child's brain and kidney, and even minimal exposure can affect its learning abilities.

Greenhouse gases lead to warmer temperatures, by trapping the earth's heat in the atmosphere, which in turn lead to what is known as climate change. Rising sea levels, higher chance of extreme weather, heat-related deaths, and the increased transmission of infectious diseases are all by-products of those pollutants. Some of the worst are hydrofluorocarbons (HFCs) that can be found in air conditioners and refrigerators, are thousands of times more powerful in trapping heat than carbon dioxide. In October 2016 more than 140 countries reached an agreement to cut back the use of such chemicals and develop greener alternatives. Though the President of the United States of America was unwilling to sign on to this agreement, a two-party system of senators overrode his objection in 2020. Therefore, the United States is on track to slash HFCs by 85 percent till 2035.

Climate change can exacerbate the effect of other particles on humans. For instance, trees, weeds, and grass can produce allergens that are also carried in the air and can be hazardous to health. Though they are less directly connected to human actions, they can be treated as a form of air pollution. Climate change also lengthens the pollen production season, and some studies have noticed that ragweed pollen itself might be becoming a more potent allergen.

According to the most recent State of Global air report 2.2 million deaths were caused by indoor air pollution and 4.5 million deaths were linked to outdoor air pollution exposures in 2019. Air pollution is now the world's 4^{th} largest risk factor for early death [3].

1.2 Combating Air Pollution

Each nation has taken actions to combat air pollution according to specific standards set by authorized organizations.

On 31 December 2016 in Europe, a reductions directive named National Emissions Commitments (NEC) entered into force which sets 2020 and 2030 emission reduction commitments for five main air pollutants: NO_x, NMVOCs $SO_{2,}$ $NH_{3,}$ $PM_{2.5,}$ CO.

In addition, PM_{10} (particulate matter), if available CB (black carbon), TSP (total suspend particulate matter), Cd (heavy metals cadmium), Pb (lead), Hg (mercury), PAHs (polycyclic aromatic hydrocarbons), dioxins and furans are taking into consideration. The European Environment Agency (EEA) compiles a yearly status report supported by its European Topic Centre (ETC) and checks the data reported by the members under the NEC Directive. Furthermore, an annual status briefing is prepared that summarizes the most recent reported data and progress of Member States and the EU in meeting their emission ceiling obligation [4].

In the United States, the Environmental Protection Agency (EPA) has taken significant steps to dramatically reduce toxic air pollutants and provide important health protections nationwide. These include reducing toxic emissions from industrial sources, from vehicles and engines through new stringent emission standards and cleaner-burning gasoline and, addressing indoor air pollution through voluntary programs.

The Clean Air Act (CAA) requires regulations of air pollutants that are hazardous from large industrial facilities. It achieves that through two phases.

The first phase called "technology-based," where the EPA develops standards for controlling the air emissions from sources in an industry group. The maximum achievable control technology standards (MACT) are based on the industry's emissions levels and are being regulated by the emitting sources.

Within 8 years of setting the MACT standards, the second phase "risk-based" approach called residual risk has the CAA direct the EPA to assess the remaining health risks from each category to determine if the MACT standards protect public health with a great emphasis on safety against adverse environmental effects. Not to mention that every 8 years after the MACT standards have been set, the CAA demands the EPA to review, modify them, and if required, to account for improvements in air pollution prevention and/or controls.

Another administrative program that reduced mobile source air pollutants was the 2007 mobile source air toxics rule, which aimed to control the benzene content of gasoline and vehicle emissions at cold temperatures.

It can be determined that the CAA has been a critical tool for reducing air pollution since its passage in 1970, although fossil-fuel interests aided by industry-friendly lawmakers have frequently attempted to weaken its many protections. Ensuring that such environmental laws remain intact and enforced properly will always be the key to improving and maintaining air quality [5].

In Asia, matters are much more complicated. Most of Asia and thus the Pacific's population are exposed to levels of air pollution that pose a major risk to their health, on the environment, and agricultural crop yields. These impacts have serious economic consequences, affecting economic growth and welfare. While current laws and policies seem to have made progress in reducing air pollution within the region, there are still further actions that need to be taken to bring air quality to safe levels. The most damaging air pollutants are fine Particulate Matter ($PM_{2.5}$) and Ground-level Ozone (O_3). While the sources of these pollutants differ from country to country, they are only linked with a handful of activities. Urban and heavily industrialized areas have the highest levels of air pollution, especially with high population density.

Things seem worrying and there is no clear image of what is truly going on. For instance, a report "Air Pollution in Asia and the Pacific: Science-based solutions" [6] that was published in 2019 claims that things are taking a negative direction while other articles from Earth.org [7] for example claim that China has made strides in combating air pollution. More specifically, it claims that the Chinese government has introduced afforestation and reforestation programs like the "Great Green Wall" and planted more than 35 billion trees across many provinces. China's forestry investments of over $100 billion per hectare exceeded those of the US and Europe and became three times higher than the global average. In addition, it claims that from 2013 to 2017 there was 33% reduction of $PM_{2.5}$ in Beijing. Still, it is too soon to come to a conclusion about what is really happening.

1.3 Air Quality Index

The Air Quality Index (AQI) is an index for reporting periodic air quality. It is used as an indicator of how clean or polluted the air is for the general public, and what associated health effects should be of concern. Consider the AQI as an indicator that goes from 0 to 100. The higher the AQI value, the greater the correlation of air pollution and health concern. For example, an AQI value of 80 or higher represents hazardous air quality conditions, while a value of 10 or even lower represents good air quality. During a period of poor air quality, when the AQI indicates that prolonged exposure may cause significant harm to the public health, then agencies may resort to emergency plans that allow them to order emitters like factories to curtail emissions until the hazardous conditions abate and return to acceptable levels.

In 1968 the National Air Pollution Control Administration (NAPCA) launched an initiative to develop a measuring methodology called Air Quality Index and to apply it to Metropolitan Statistical Areas (MSA). The incentive was to draw public attention to the issue of air pollution and indirectly push local public officials to take action to control sources of pollution and enhance air quality within their administration. The original iteration generated individual pollutant indices by using standardized ambient pollutant concentrations. These indices were then weighted and summed to create a single total AQI. The methodology would use concentrations that are taken from monitoring data or are predicted by means of a diffusion model. Afterwards, they were converted into a statistical distribution with a standard deviation and preset mean. The methodology was designed to be robust however, the practical application for all metropolitan areas proved to be inconsistent due to the paucity of ambient air quality monitoring data, lack of agreement on weighting factors, and dissimilarity of air quality standards across geographical and political boundaries. Today different countries and nations have their own AQIs, corresponding to different national air quality standards [8][9].

In Europe the European Air Quality Index allows its users to be informed about the air quality in their place of residence, work, or travel. It is based on concentration values for 5 key pollutants, including:

- fine particulate matter ($PM_{2.5}$)
- particulate matter (PM_{10})
- nitrogen dioxide (NO_2)
- ozone (O_3)

- sulphur dioxide (SO_2)

The index is computed hourly from more than two thousand air quality monitoring stations across Europe, using up-to-date data reported by EEA member countries. The concentrations values for 5 key pollutants determine the index level that reflects air quality. The index corresponds to the poorest level for any of the 5 pollutants, according to the table below.

	Good	Fair	Moderate	Poor	Very poor	Extremely poor
$PM_{2.5}$	0-10	10-20	20-25	25-50	50-75	75-800
PM_{10}	0-20	20-40	40-50	50-100	100-150	150-1200
NO_2	0-40	40-90	90-120	120-230	230-340	340-1000
O_3	0-50	50-100	100-130	130-240	240-380	380-800
SO_2	0-100	100-200	200-350	350-500	500-750	750-1250

Table 1. Air Quality Index table for Europe

Each one comes complemented with health-related messages:

AQ index	General population	Sensitive populations
Good	The air quality is good. Enjoy your usual outdoor activities.	The air quality is good. Enjoy your usual outdoor activities.
Fair	Enjoy your usual outdoor activities	Enjoy your usual outdoor activities
Moderate	Enjoy your usual outdoor activities	Consider reducing intense outdoor activities if you experience symptoms.
Poor	Consider reducing intense activities outdoors, if you experience symptoms such as sore eyes, a cough or sore throat	Consider reducing physical activities, particularly outdoors, especially if you experience symptoms.
Very poor	Consider reducing intense activities outdoors, if you experience symptoms such as sore eyes, a cough or sore throat	Reduce physical activities, particularly outdoors, especially if you experience symptoms.
Extremely poor	Reduce physical activities outdoors.	Avoid physical activities outdoors.

Table 2. Messages for each color indication of the Air Quality Index table of Europe

The messages are based on the relative risks associated with short-term exposure to $PM_{2.5}$, O_3, and NO_2, as defined by the World Health Organization in its report on the Health Risks of Air Pollution in Europe project (HRAPIE project report) [10].

- The relative risk of exposure to $PM_{2.5}$ is taken as a basis for driving the index, specifically the increase in the risk of mortality per 10 μg/m3 increase in the daily mean concentration of $PM_{2.5}$.
- Assuming linearity across the relative risk's functions for O_3 and NO_2, we calculate the concentrations of these pollutants that pose an equivalent relative risk to a 10 μg/m3 increase in the daily mean of $PM_{2.5}$.
- For PM_{10}, a constant ratio between PM_{10} and $PM_{2.5}$ of 1:2 is assumed, in line with the World Health Organization´s air quality guidelines for Europe.
- For SO_2, the bands indicate the value's limits set from the EU Air Quality Directive.

In the United States, the Environmental Protection Agency (EPA) has developed an AQI which is split into six categories indicating different health concern levels. It takes into consideration all the criteria air pollutants measured within a geographic area.

The EPA has established a national air quality index since 1976 to provide a daily report on air quality that is easy to understand in a format that's identical from state to state.

Since then, it has been updated to reflect the latest health-based air quality standards several times.

There is a U.S. AQI for five pollutants that are being regulated by the CAA:

- particulate matter ($PM_{2.5}$)
- carbon monoxide (CO)
- nitrogen dioxide (NO_2)
- ozone (O_3)
- sulfur dioxide (SO_2)

The AQI is based on the national health-related air quality standard for each pollutant and the scientific information that reinforces that it individually. The timeframe differs by pollutant. The O_3 AQI is an 8-hour index, while for particle pollution and the other it's 24 hours.

Below is a table showcasing each pollutant and how it should be interpreted by the public:

AQI Color	Levels	Values of Index	Description
Green	Good	0 - 50	Air pollution poses little or no risk.
Yellow	Moderate	51 - 100	There may be a risk for some people, particularly those who are unusually sensitive to air pollution.
Orange	Unhealthy for sensitive groups	101 - 150	Sensitive groups may experience health effects. The public is less likely to be affected.
Red	Unhealthy	151 - 200	Members of the public may experience health effects, sensitive groups may experience serious health effects.
Purple	Very unhealthy	201 - 300	The risk of health effects is increased for everyone.
Maroon	Hazardous	301 and higher	This is an emergency conditions where everyone is more likely to be affected.

Table 3. Air Quality Index table of United States of America

Each category correlates to a different level of health concern and a specific color. It is used for common and scientific people to determine whether air quality is reaching dangerous levels of health concern in their communities and what groups it may affect or if any [11][12].

In Asia, things again are more complicated. Most countries and even some cities have their own AQI measurements and systems that were established from their own policies and authorities. In addition, different groups of pollutants are considered as harmful for each. Below 4 locations are given as examples to showcase how each place handles the problem.

In mainland China their ministry of Environmental Protection is responsible for measuring the level of six atmospheric pollutants to determine the state of air pollution:

- particulate matter ($PM_{2.5}$)
- particulate matter (PM_{10})
- carbon monoxide (CO)
- nitrogen dioxide (NO_2)
- ozone (O_3)
- sulfur dioxide (SO_2)

To each pollutant, an Individual Air Quality Index (IAQI) is appointed to and afterward, the final AQI is the highest value of the above. The final AQI value can be calculated either every 1 or 24 hours.

Each pollutant's concentrations are measured quite differently. When the AQI is calculated every 1 hour, then the SO_2, NO_2, and CO are calculated as an average for every 24 hours, while the O_3 concentration is measured as average every 1 hour and the moving 8-hour average, and the $PM_{2.5}$, PM_{10} concentrations are measured as an average either every 1 or 24 hours.

However, when the AQI is calculated every 24 hours, then an average from the SO_2, NO_2, CO, $PM_{2.5}$, and PM_{10} is measured every 24 hours, while the O_3 concentration is measured as the maximum 1-hour average and the maximum moving average of every 24th hour.

Each pollutant's IAQI is calculated in consonance with a formula published by the MEP and the resulting value is non-linear for each, as is the final AQI. Therefore, it should be clarified that an AQI of 200 should not be taken as twice the pollution of an AQI value 100 nor that the air is twice as harmful. If the AQI of day 1 to day 150 goes from 20 to 180 the average would be 100 which lies in a safe benchmark.

Still, the pollution is considered unacceptable due to the way that the AQI changed due to the time range that it was measured.

Because the reference point is a 24-hour target, and the annual target's value must match the annual average there is a great chance to have safe air every day of the year but still fail the annual pollution reference point.

Below is a table showcasing how the public should act according to the level of the AQI and the health implications it comes with. Note that the table holds different statements when it comes to the pollutant of $PM_{2.5}$:

AQI	**Levels**	**Health Implications**	**Statement (for PM2.5)**
0 - 50	Good	Air pollution poses little or no risk	None
51 -100	Moderate	For some pollutants there may be a moderate health concern for a very small number of people who are unusually sensitive to air pollution.	Active children and adults, and people with respiratory disease, such as asthma, should limit prolonged outdoor exertion.
101-150	Unhealthy for Sensitive Groups	Members of sensitive groups may experience health effects.	Active children and adults, and people with respiratory disease, such as asthma, should limit prolonged outdoor exertion.
151-200	Unhealthy	Everyone may begin to experience health effects, members of sensitive groups may experience more serious health effects	Active children and adults, and people with respiratory disease, should avoid prolonged outdoor exertion, everyone else, especially children, should limit prolonged outdoor exertion
201-300	Very Unhealthy	The entire population is more likely to be affected.	Active children and adults, and people with respiratory disease, should avoid all outdoor exertion; everyone else, especially children, should limit outdoor exertion.
300+	Hazardous	everyone may experience more serious health effects	Everyone should avoid all outdoor exertion

Table 4. Messages for each color indication of the Air Quality Index table of United States of America

In China's Hong Kong uses a different Index which is measured of 1-10+ and considers only four atmospheric pollutants:

- $PM_{2.5}$ (particulate matter)
- NO_2 (nitrogen dioxide)
- O_3 (ozone)
- SO_2 (sulfur dioxide)

The AQHI is calculated from the sum of the percentage of hospital admissions with health-related concerns on a day-to-day basis while being attributable to the 3-hour moving average concentrations of these 6 pollutants.

Health Risk Category	**AQHI**
Low	1 - 3
Moderate	4 – 6
High	7
Very high	8 – 10
Serious	10+

Table 5. Air Quality Index table of Hong Kong

On the other hand, India's Central Pollution Control Board along with State Pollution Control Boards has been operating the National Air Monitoring Program for its AQI. A group of experts comprised of medical professionals, air quality experts, academia, and SPCBs have developed an AQI that calculates 8 air pollutants:

- particulate matter ($PM_{2.5}$)
- particulate matter (PM_{10})
- carbon monoxide (CO)
- nitrogen dioxide (NO_2)
- ozone (O_3)
- sulfur dioxide (SO_2)
- Ammonia (NH_3)
- Lead (Pb)

Each one is prescribed a short-term (1 to 24 hourly average period) National Ambient Air Quality Standard. Based on the measured concentrations, an index is calculated for each of the pollutants, corresponding to standards and possible health impact. The worst index reflects overall AQI.

Below is a table with the health impacts in accordance with the AQI:

AQI	Remark	Health Impacts
0 - 50	Good	Minimal Impact
51 – 100	Satisfactory	Minor breathing discomfort to sensitive people
101 – 200	Moderate	Breathing discomfort to the people with lungs, asthma, and heart diseases
201 – 300	Poor	Breathing discomfort to most people on prolonged exposure
301 – 400	Very Poor	Respiratory illness on prolonged exposure
401 – 500	Severe	Affects healthy people and seriously impacts those with existing diseases

Table 6. Air Quality Index table of India

To make matters even more confusing, even in South Korea their official organizations like the Ministry of Environment uses the Comprehensive Air-quality index (CAI) to describe air quality based on the health risks of air pollution which is completely different from the others. It is measured from 0 to 500 value and is divided into 4 categories indicating the level of air pollution in the specific region. The highest value of the 5 major pollutants is equivalent to the CAI value. The index has also correlated health effects with a color representation of the categories as shown in the table below.

CAI	Description	Health Implications
0–50	Good	A level that will not impact patients suffering from diseases related to air pollution.
51–100	Moderate	A level that may have a meager impact on patients in case of chronic exposure.
101–250	Unhealthy	A level that may have harmful impacts on patients and members of sensitive groups and cause the public unpleasant feelings.
251–500	Very unhealthy	A level that may have a serious impact on patients and members of sensitive groups in case of acute exposure.

Table 7. Air Quality Index table of South Korea

To summarize, each country in Asia is trying to solve air pollution in their own way with their own programs and groups. However, this makes matters difficult when other officials try to see a general image of what is really happening [8].

2. Forecasting Air Pollution

Air pollution forecasting is considered a worthwhile investment on multiple levels for both individual, community, national and global. Accurate forecasting helps people to prepare for the worst while decreasing the effects on public health and safety.

If people are aware of variations in the quality of the air, they will be able to protect themselves from the effect of pollutants on their health as well as concentrations likely to cause adverse effects in the long run. Furthermore, there is a greater chance of motivating changes in both individual behavior and public policy, since people will want to have more, accurate air quality information and better air quality. Such awareness has the potential to invoke actions for a cleaner environment and a healthier population. Not to mention the fact that Governments can benefit from forecasting air pollution because they can legislate procedures that will combat the severity of pollution levels.

When an expert tries predicting air quality, there are many factors and variables that need to be taken into account, some of which are fairly unpredictable. For example, Beijing's authorities sometimes order coal plants, facilities, and factories to seize operations and ban a portion of the city's registered vehicles from circulating. Natural disasters are also another unpredictable variable. In Greece, a great culprit of its poor air quality is the constant wildfires which occur almost every summer.

Therefore, the complications that follow weather forecasting are involved in predicting air quality but also it requires diverse data and knowledge of:

- Local pollutant concentrations and emissions
- Pollutant emissions from other regions and distant locations
- Movements and possible transformations of pollutants
- Prevailing winds

To sum it up, the many subjective and objective factors at play in predicting valid air quality results in air pollution forecasting.

There are many forecasting techniques and methodologies, and most of them require more complexity than standard weather forecast models. These models are mathematical simulations of how airborne pollutants dissipate in the air. Meteorological forecasting is the practice and application of such models and brings with it many approaches to the problem or even combinations of them all [13]. A few examples are:

- **Climatology** assumes that the past is a good indicator of the future. It is based on the relationship between specific weather conditions and pollution levels and consequently, can be very one-dimensional. Usually, it is extended to include the correspondence of weather to pollution patterns. Still, there are many limitations to this method, and it is mostly used as a tool to support other forecasting methods.
- **Classification and regression tree** is designed to classify data into dissimilar groups. The algorithm identifies variables that correspond to pollution levels by using data to forecast concentrations in relation to weather conditions and correlated pollutant concentrations.
- **Regression analysis** estimates relationships between variables by analyzing historical data sets, their associations between pollution levels and, meteorological data variables to result in an equation that can be used to forecast future pollution levels.
- **Neural networks** use complicated, performance-intensive, adaptive learning and pattern recognition techniques. Computer-based algorithms are designed to simulate the human brain's capability in order to counter such complicated problems. It is considered one of the most suited methods for forecasting pollution due to its multi-dimensional approach.

In this thesis, the application of Neural Networks will be used to forecast the future values for the 5 main pollutants of the AQI of Europe.

3. Time Series Forecasting

Over the years many approaches have been taken to forecast time series with Machine Learning and Neural Networks. The way forecasting works is by using historical data to fit on models and using those models to predict future observations in reference to them. An important distinction in other methodologies and forecasting is that future values must never be available and only be estimated from what has already happened in the past. The performance of such a model is determined by its ability to predict the future correctly or come very close to the actual value. This is often at the cost of being able to explain why a prediction was made, by understanding the causes that are underlying the problem.

The way time series forecasting is handled with machine learning is that it is framed as a supervised learning problem. This re-framing of time series data allows the usage of machine learning algorithms both standard linear and nonlinear to tackle the problem. Supervised learning at its core is where we have input (X) and, output variables (y) and apply an algorithm to learn the mapping function and relevance from one to the other. The goal is to approximate the real mapping inner workings that lie underneath so that when we have new input data (X), we can predict the output (y) for that data. It is called supervised learning because of the process of using a training dataset to make an algorithm learn from. It can be viewed as a teacher supervising the learning process of a student. Supervised learning problems can be separated into groups of regression and classification problems.

- **Classification** is when the output variable is one of many possible known categories, such as “blue” and “red” or “dog” and “cat.”
- **Regression** is when the output variable is a real value that is affected from its previous values but also other features, such as “money” or “weight.”

In the case of air pollution forecasting the problem will be approached as a regression problem. Noted that the number of observations recorded for a given time in a time series dataset matters because they are handled differently. They are split into 2 types:

- **Univariate Time Series:** Datasets where one variable is only used as an observation at each time, for example, the temperature for each hour.
- **Multivariate Time Series**: Datasets where more than one variable is observed at each time, for example, the humidity and temperature for each hour.

Using machine learning for time series forecasting is where classical methods fall, and it comes with many approaches to the matter. The model types that are used today are:

- **Multi-layer Perceptron (MLP)**: is a class of feedforward artificial neural networks that subsists of at least 3 layers (input, hidden, output) of nodes and utilizes backpropagation for training.
- **Convolutional Neural Network (CNN)**: is a class of artificial neural networks, that is usually applied to analyze visual imagery. They are based on a weight architecture that is shared of the convolution kernels or filters that slide along input features and produce feature maps.
- **Recurrent Neural Networks (RNN):** is a class of artificial neural networks where nodes are connected to one another in a way that composes a graph that is directed neighboring to a sequence. They can use an internal state (memory) and other mechanisms to process sequences of inputs and find out patterns and trends.

- **Long Short-Term Memory Networks (LSTM):** is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. In comparison to the standard feedforward neural networks, the LSTM has feedback connections that enable them to process long sequences more efficiently. It can process both single data points and entire sequences of data.

It should be noted that many experts claim that time series forecasting is difficult since time series problems add the complexity of order or temporal dependence between observations.

Appropriate data handling is required when fitting and evaluating models in order to avoid reaching false positives. It also aids in modeling, providing additional structure like trends and seasonality that can be leveraged to improve model skill [14][15][16].

Still, the use of machine learning comes with 2 major issues when it comes to time series forecasting namely Autocorrelation and Prediction Uncertainty:

Autocorrelation is a mathematical representation for the degree of similarity between the predicted and actual time series values. Over successive time intervals if a lagged version of the predicted values appear after the change of the actual values, then that means the predictions of the model are inaccurate and highly determined by the current past. A representation of +1 in the time steps means a perfect positive correlation, while a -1 is a negative correlation [17].

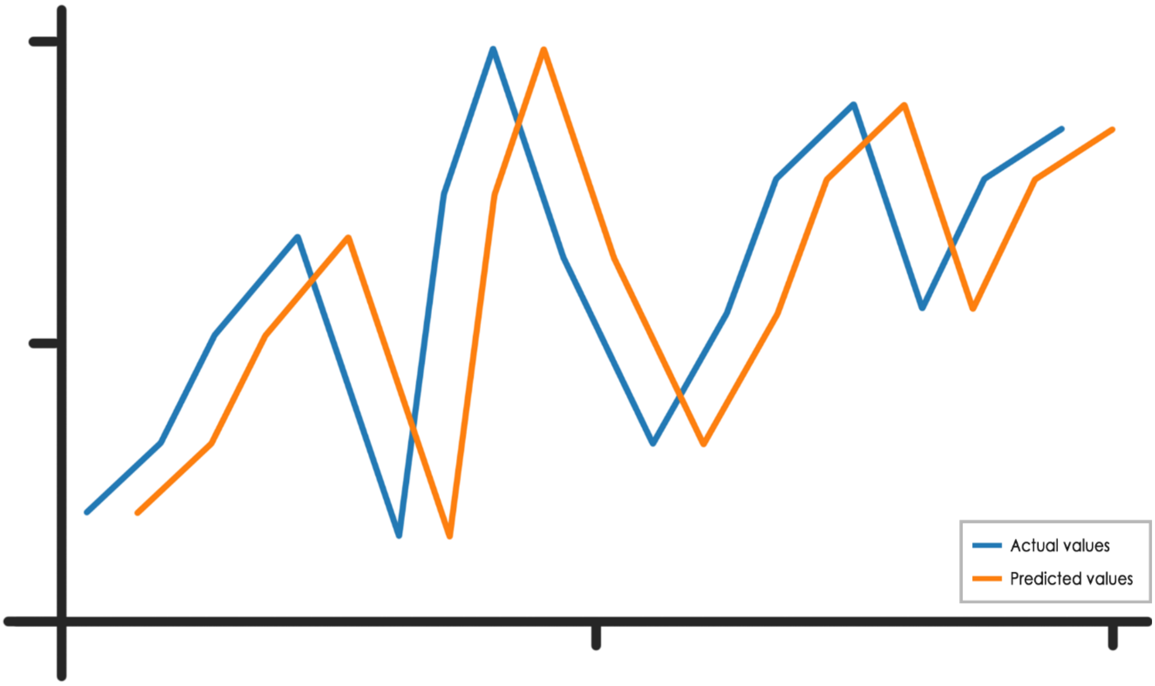


Figure 2. An example of Autocorrelation in a time series

Uncertainty quantification [18][19] is a concept in machine learning that tries to answer what can a model know and not know in accordance with the data it has been provided as its inputs. There are 2 major types:

- **Aleatoric uncertainty** also known as data uncertainty is when no matter how much data is collected, there is still a possibility the prediction of the model will be different from the actual value. The subway problem is a great illustration of how this type of uncertainty works. Let's say that we try to predict the schedule of several trains and when they arrive. No matter what parameters are provided to the model there is still a possibility where it will return a wrong prediction. Situations, where a train gets damaged or is maintained, are parameters that are unknown to the model and as a result, affect the accuracy of its prediction.
- **Epistemic uncertainty** also known as model uncertainty is when the parameters of the model are ignorant of situations that might be predictable. This can be reduced as more data is provided. The coconut problem is a great way of showcasing how such a problem is handled. Let's say that we have someone that stands below coconut trees, and we want to predict when a coconut will fall on their heads. Even though we have provided data that hold information about the weather or the age of each tree there are still other features that can be added such as the moments when the person is not under a coconut tree. As a result, we can reduce the window of uncertainty and be able to have better accuracy of the model.

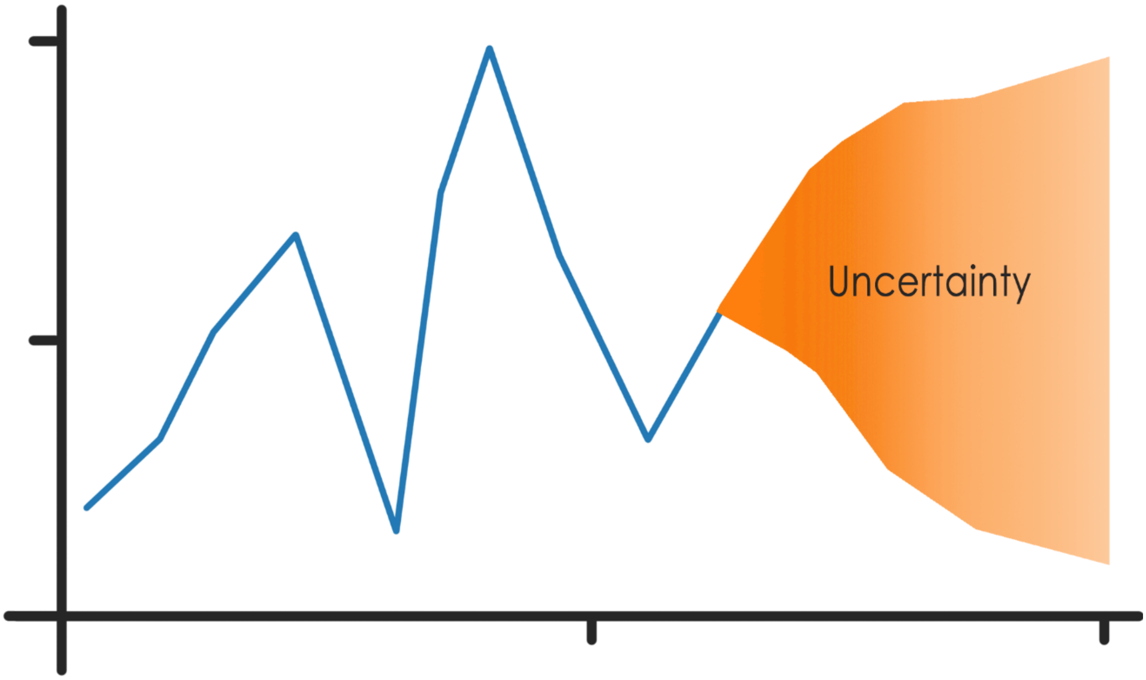


Figure 3. Uncertainty of a model's prediction

4. Artificial Neural Networks

4.1 Introduction

Artificial Neural networks (ANN) are Machine Learning models that were inspired by the inner workings of biological neurons found in the human brain. They are very versatile, and scalable, making them ideal to tackle highly complex Machine Learning tasks such as classifying billions of images, speech recognition services, recommendation systems, or even learn to play the game of Go.

The history of ANNs started back in 1943 by the neurophysiologist Warren McCulloch and the mathematician Walter Pitts. They proposed a computational model that showcased how neurons could work with one another to perform complex computations using mathematical and propositional logic. Thus, the first artificial neural network architecture was created.

Today, ANNs frequently outperforms ML techniques on large and complex problems especially with the abundance of available data to train them. Furthermore, ANNs have entered a period of funding and progress. Products based on ANNs regularly make the news, which attracts more attention and funding toward them, resulting in more progress and products [20].

4.2 Computations inside the perceptron

A perceptron is a unit of a neural network that can capture features from the input data by doing certain computations. It is one of the simplest ANN architectures that takes inputs and outputs as numbers and a weight is associated with each input connection. It computes a weighted sum of its inputs and then applies a transfer/step function to that sum and returns an output. Moreover, a bias feature is given in the transfer function to affect the accuracy of the output.

$$f(x) = \begin{cases} 1 \ \ if \ w \cdot x + b > 0, \\ 0 \qquad \quad otherwise \end{cases}$$

Where $w =$ vector weights, $w \ \cdot \ x$ is the $\sum_{i=1}^{m} w_i \cdot \ x_i$ where m is the number of inputs and b is the bias which does not depend on any input value and shifts the decision boundary away from the origin [21].

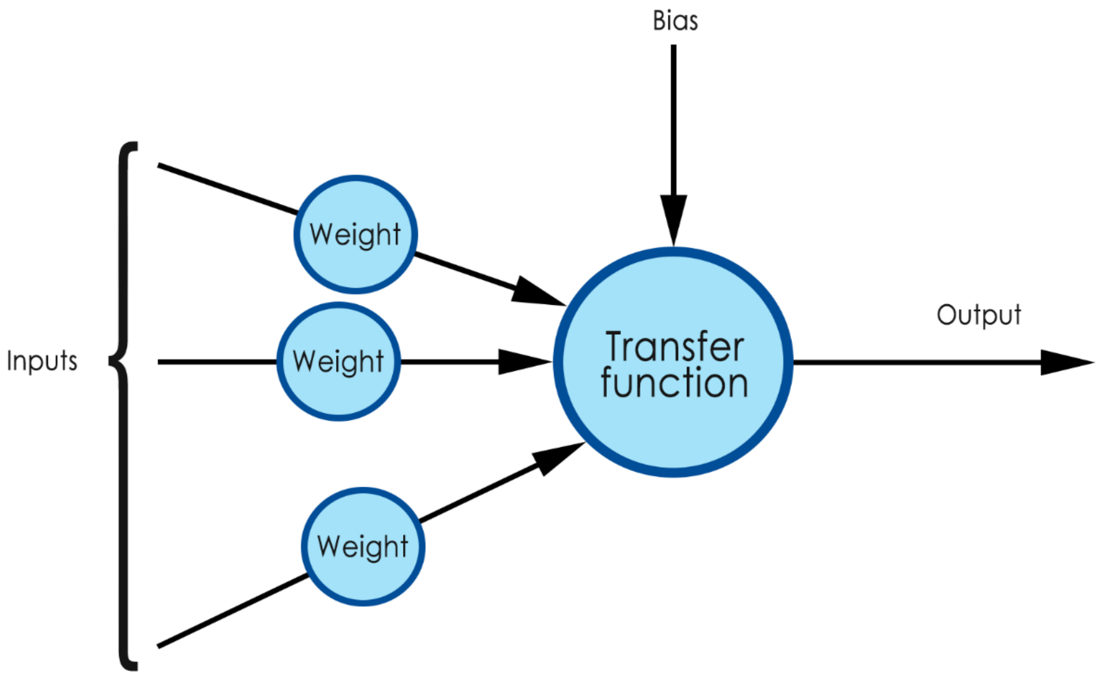


Figure 4. The architecture of a Perceptron

The perceptron can be used for simple linear binary classification problems. It calculates a linear combination of the inputs, and according to the result, it outputs a positive or negative class. The perceptron on each own is composed of a single layer threshold logic unit with each one connected to all the inputs. When all neurons in a layer are connected to all the neurons of the previous layer then that layer is called fully connected or dense layer.

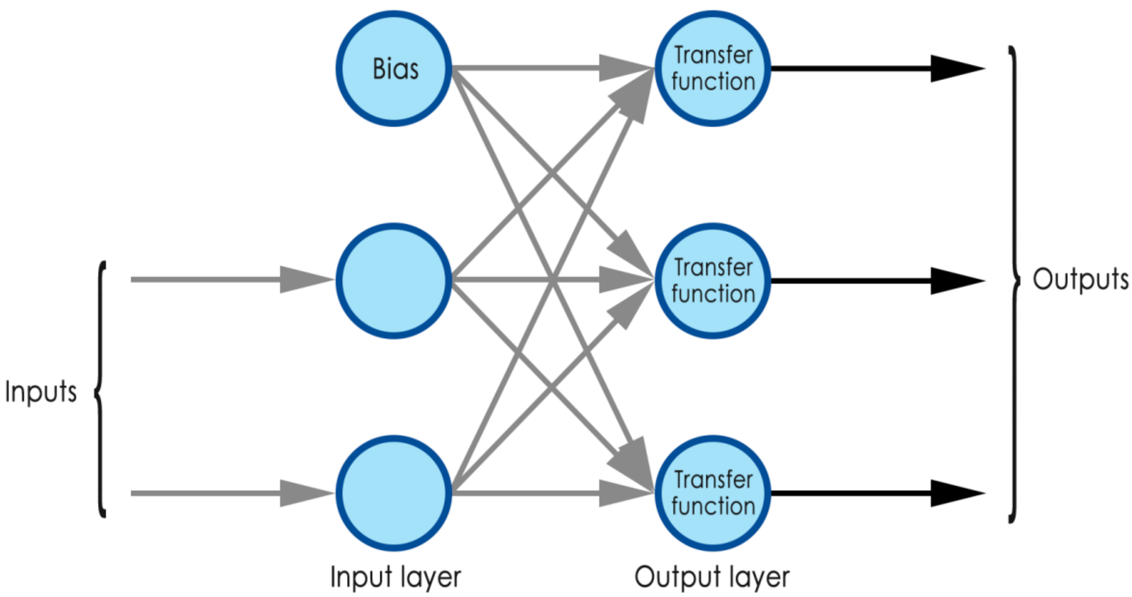


Figure 5. The architecture of a Neural Network layer (Dense layer)

As a result, it makes it possible to compute outputs of a layer of artificial neurons for several instances at once. The training of a perceptron and of any neural network is achieved by taking into account the error made by the network when it makes a prediction and reinforces connections that help reduce the error. More specifically, it reinforces the connection weights from the inputs that have contributed towards the correct prediction, while for every output neuron that returned a wrong prediction it weakens them. Such a concept can be taken a step further with the Multilayer Perceptron (MLP) where there would be an input layer, one or more hidden layers, and an output layer that solve more trivial problems more effectively.

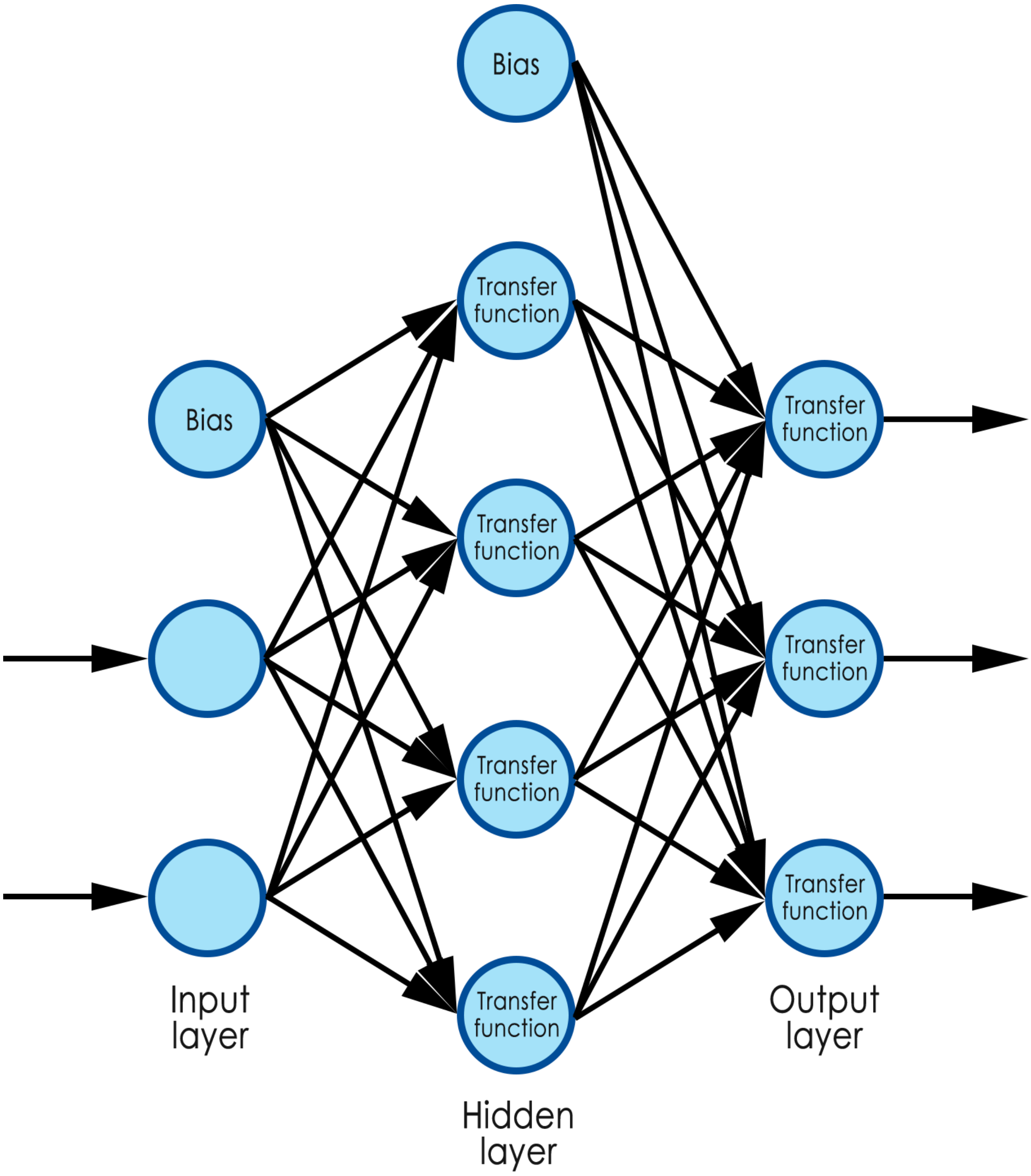


Figure 6. The architecture of a Multi-Layer Perceptron

Furthermore, by using a backpropagation training algorithm, it can find out how each connection weight and each bias value should be changed to reduce the error. The way it works is that for each training instance the algorithm makes a prediction and compares it with the error, then goes through each layer to find how each connection contributed to the error and adjusts the connection weight to reduce it. MLPs can be used for regression and classification tasks however they too have their own limits.

It should also be noted that such models are also called Feed-forward neural networks because the information that enters only moves in one direction — from the input layer, through the hidden layers, to the output layer. In other words, the information moves straight through the network and never touches a node twice [22][23].

Today there is an abundance of NN architectures that have been developed to tackle more complex problems and many fields of study have benefited from it. Some of them are Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs).

4.3 One-dimensional Convolutional Neural Networks

There are many types of CNNs that have been developed over the years. Each is used for a different purpose for numerous fields of study such as computer vision, image recognition, and more. Their architecture is analogous to that of the connectivity pattern of Neurons in the brain, individual neurons respond to stimuli only in a restricted region of the visual field and as a collection overlap to cover the entire visual area. In other words, through the application of appropriate filters, it is able to capture the spatial and temporal dependencies in an image successfully [24].

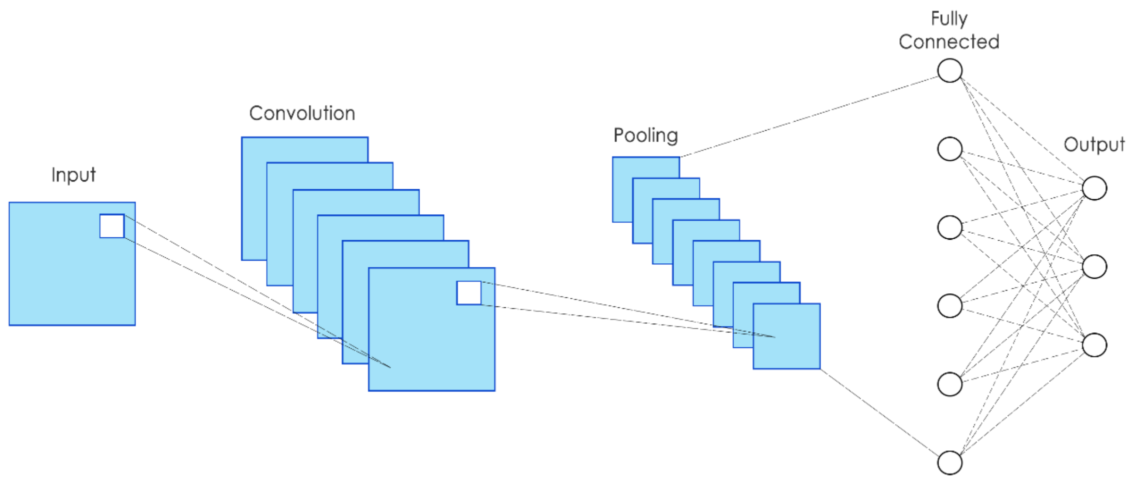


Figure 7. A standard Convolutional Neural Network's architecture

However standard CNNs are unfit for time series forecasting due to the fact that they were designed without the thought of handling one-dimensional inputs. As a result, 1-dimensional convolutional neural networks (Conv1D) were developed to tackle such problems and not to mention the fact that recent studies show they perform comparably or even better to the standard methodologies. The way it works is that it moves the kernel from left to right along with the signal at one data point at a time and measures the dot product of the two at each position. It achieves that by multiplying each pair together and adding up those products. This results in a convolution consisting of a sequence of dot products from the signal with the kernel.

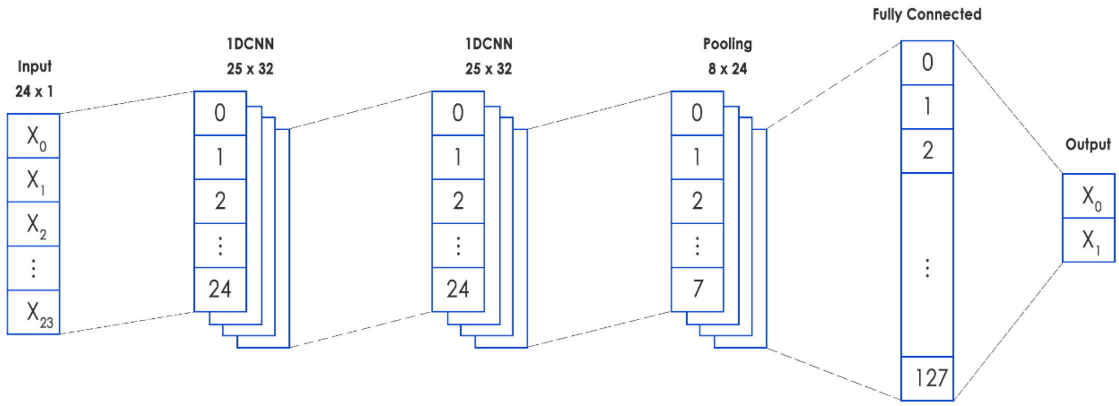


Figure 8. A 1-dimensional Convolutional Neural Network in a model

The following equations measure result y of the Conv1d:

$$y_t = \sum_{k=-p}^{p} x_{j-k} \cdot w_k$$

Where x is the signal, w is the kernel, and p is half of the length of the signal.

Conv1D consists of filters that are applied to the inputs and features shape, a kernel size that is the size of the sequential window of the input, and a dilation/ skipping rate for the kernels [25][26].

4.4 Recurrent Neural Networks

The concept of recurrent neural networks (RNNs) is that in contrast to Feed-forward neural networks the information cycles through a loop. When it's time to decide, it takes into consideration the current input with what it has learned from the previous inputs it received through its internal memory. In other words, RNNs add the immediate past to the present. Therefore, it has two inputs: the present and the past which are very useful when it comes to sequential data.

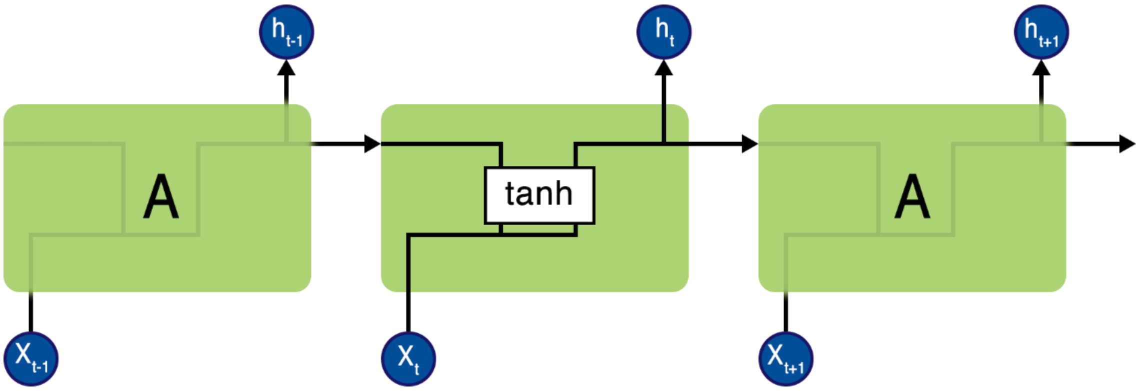


Figure 9. Architecture of a Recurrent Neural Network

It is calculated with the following equations:

$$a_t = b + W \cdot H_{t-1} + U \cdot X_t$$

$$H_t = \tanh(A_t)$$

$$O_t = C + V \cdot H_t$$

$$\widetilde{Y}_t = softmax(O_t)$$

Where α_t is the activation, b, c are the bias vectors, U, V, W are the weight matrices, H_t is the hidden state, O_t is the output state, and Y_t the actual output.

There are 4 main types of RNNs each for their own application:

- **One-to-one** is the basic and traditional type giving a single output for a single input.
- **One-to-many** is applied in situations where a single input gives a multiple-output. One of its applications would be music generation where RNN models are used to generate a music piece from a single musical note.
- **Many-to-one** is used when multiple inputs are required to give a single output. An example could be for sentiment analysis model for movie ratings where it takes review texts as input to return a rating to a movie that may range from 1 to 10.

- **Many-to-many** is as the name implies, multiple inputs and multiple outputs however it can be two kinds. One where the inputs are equal to the outputs or that each input has an output and another where they are unequal. Its case has its own application. For example, inputs=outputs can be found in Named-Entity Recognition and, inputs!=outputs can be seen in Machine translation for foreign languages.

Still, standard RNNs come with two major issues that need to be tackled. One is exploding gradients which means that the algorithm for no reason, assigns high importance to weights resulting in bad performance of the model and the other is vanishing gradient where the model stops learning because the values have become too small.

Fortunately, the concept of the Long Short-Term Memory (LSTM) unit is able to counter such issues [27][28][29].

4.5 Long Short-Term Memory

In chapter 6, a broad idea was given of the LSTM and how it works on the outside. Its internal mechanism follows a specific pattern to achieve its effects on processing sequences.

First of all, the difference between RNNs and LSTMs should be clarified. RNNs have feedback loops that let them maintain information while LSTMs have special units in addition to the standard units. These units have a set of gates that handles the data that enter, exit, and decides the importance of each item. This architecture lets them learn long-term dependencies. In addition, LSTMs deal with vanishing and exploding gradient problems that RNNs tend to have which end up having a loss function that decays exponentially with time.

LSTM networks have been designed specifically to overcome the long-term dependency problem for sequences. At the basic level the output of an LSTM is dependent on the following things:

- The cell state (current long-term memory)
- The hidden state (output at the previous point in time)
- The Input state (data input at the current time step)

Below is an image showcasing an LSTM:

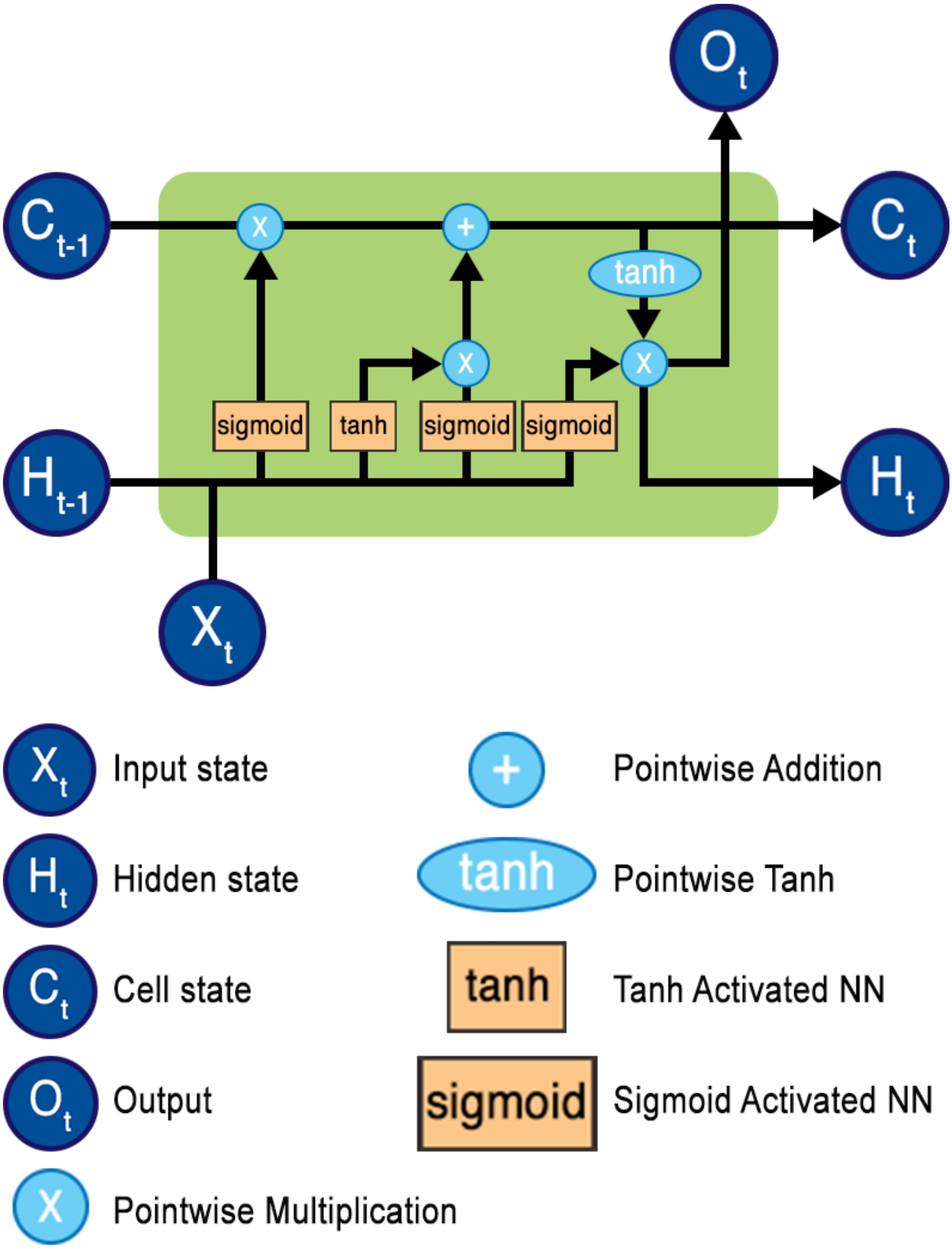


Figure 10. A Long Short-Term Memory Unit's architecture

The above visualization will be used to showcase the inner workings of the network. A standard LSTM consists of 3 gates each of which controls how the information a sequence of data is handled from the way they enter, get stored and, leave the network. They are the following:

- Input gate
- Forget gate
- Output gate

The first step is the forget gate where based on the previous hidden state and new input data we decide which parts of the cell state are useful.

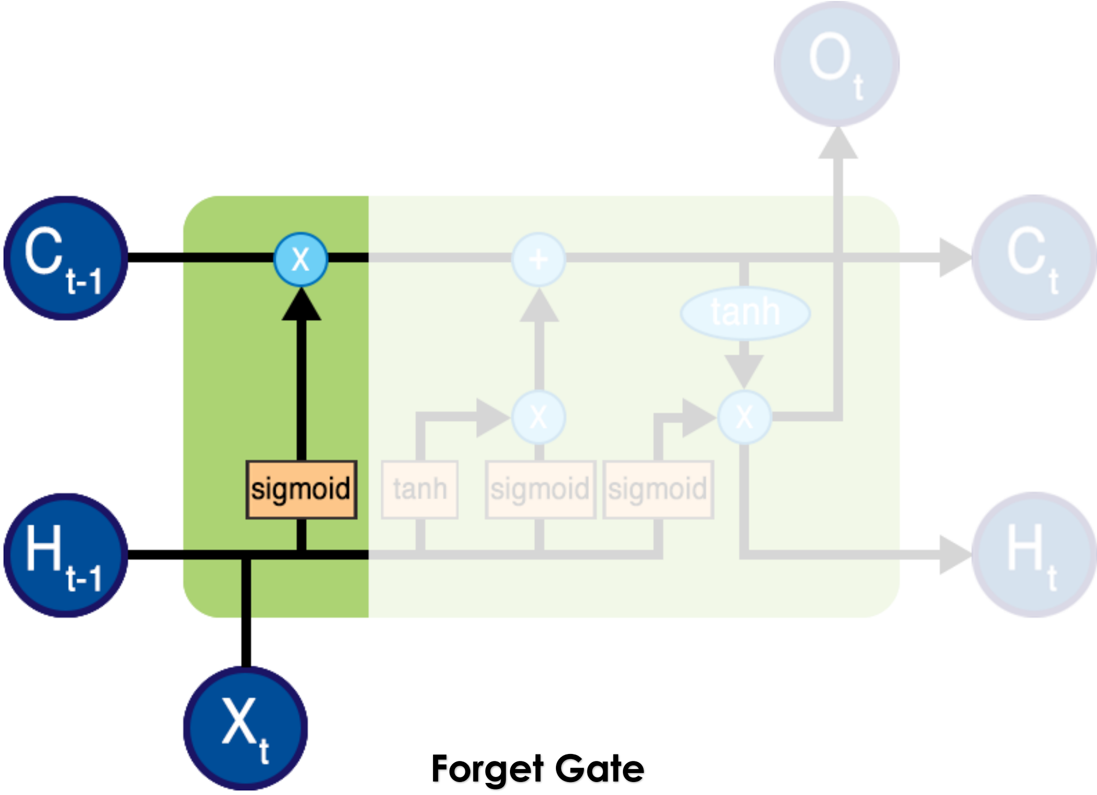


Figure 11. The Forget Gate of a Long Short-Term Memory Unit

The new input data and previous hidden state and are fed to a neural network which returns a vector in [0,1] due to the sigmoid activation. It is trained so that its output is close to 0 when a part/element of the input is regarded as irrelevant and closer to 1 when relevant. Consider each element of this vector as a kind of filter which as the value is closer to 1 it allows more information through. The output values are then sent up to the pointwise multiplied with the previous cell state. In the pointwise multiplication components of the cell state which are considered irrelevant will be multiplied by a number close to 0 and thus will have less influence on the following steps. It is calculated by the following formula:

$$f_t = \sigma(W_f \cdot [H_{t-1}, X_t] + b_f)$$

In the next step, the input gate's goal is to determine what new information should be added to the network's cell state according to the previous hidden state and input data.

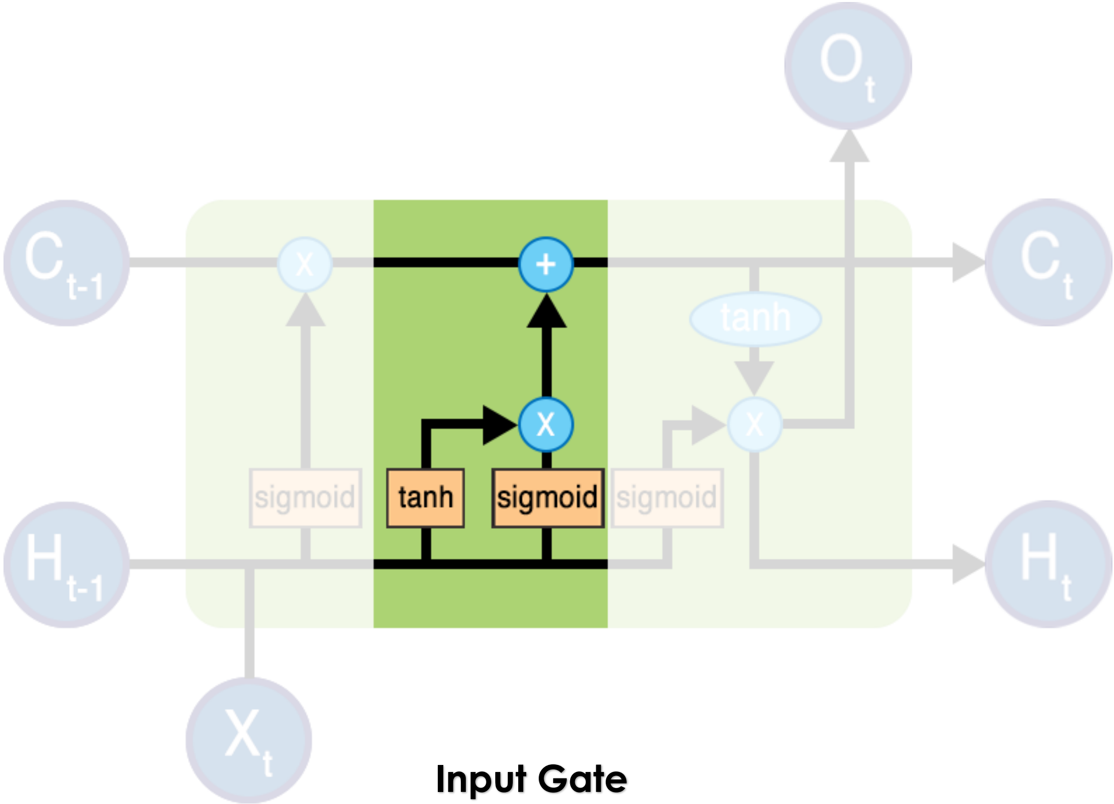


Figure 12. The Input Gate of the Long Short-Term Memory Unit

In the input gate, the tanh activated neural network (also known as a new memory network) is used to learn how to combine the previous hidden state and the input data to generate a memory update vector. This vector decides how much each component of the cell state of the network should be updated in accordance with the new data. It uses tanh which has an interval of [-1,1] which adds the possibility of negative values if it needs to reduce the impact of a component in the cell state. The formula is:

$$i_t = \sigma(W_i \cdot [H_{t-1}, X_t] + b_i)$$

$$\tilde{C}_t = tanh(W_C \cdot [H_{t-1}, X_t] + b_C)$$

Note that this part of the gate doesn't check if the input data is even worth remembering. That's why the sigmoid activated network acts as a filter, which components of the tanh activated neural network are worth keeping. Its output is a vector of values [0,1], that works as a filter from pointwise multiplication. Similarly, to the forget gate, an output near 0 is an indication for the element of the cell state if it should be updated or not.

The output of the tanh and sigmoid are pointwise multiplied which causes the new information's degree to be regulated and set to 0 if need be. Afterward, the combined vector is added to the cell state and the long-term memory is updated.

In the last step, the output gate decides the new hidden state by using the newly updated cell state, previously hidden state, and input data.

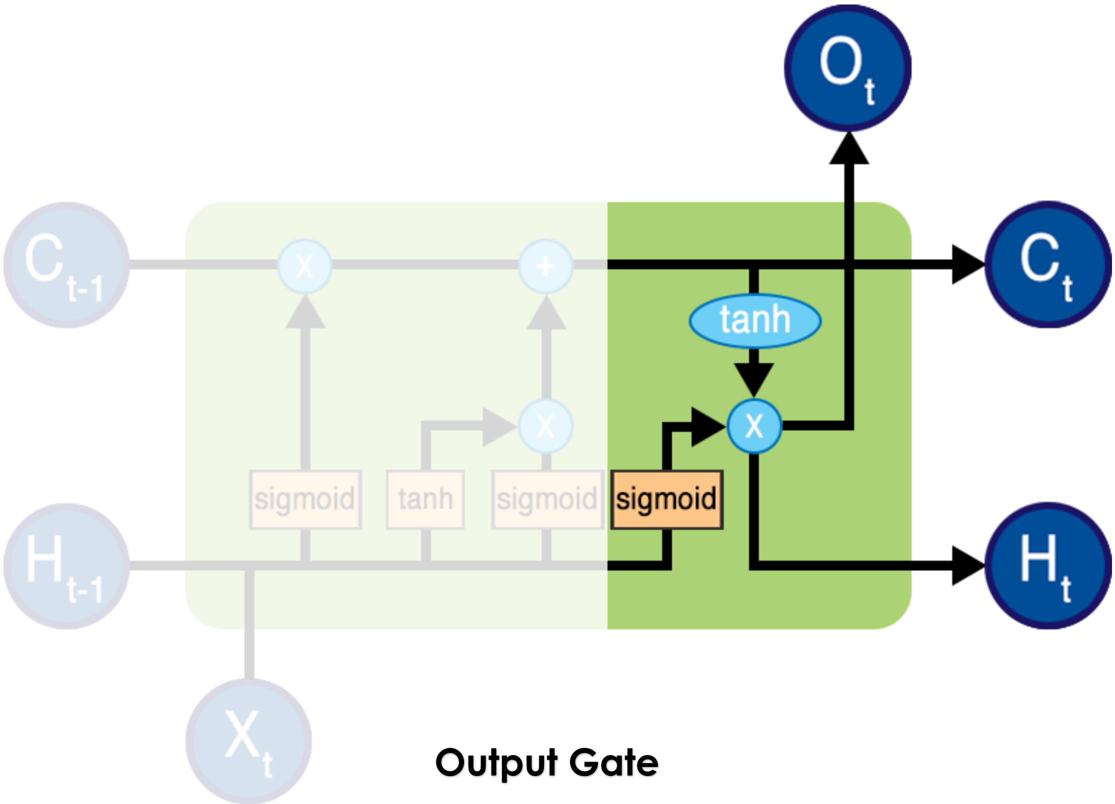


Figure 13. The Output Gate of the Long Short-Term Memory Unit

First, the tanh function is applied to the cell state pointwise that is current to acquire the new state of the cell which lies in [-1,1]. Then, the current input data and the previous hidden state go through the neural network that is sigmoid activated to access the filter vector. Afterward, the filtered vector is applied to the squished cell state by pointwise multiplication to output the new hidden state.

$$O_t = \sigma(W_O \cdot [H_{t-1}, X_t] + b_O)$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \bar{C}_t$$

$$H_t = O_t \cdot \tanh{(C_t)}$$

The above steps are repeated many times according to how far back we choose to look in the sequence. For instance, if someone wants to know what the next value of $PM_{2.5}$ will be in the next hour based on the 12 previous hours then the above steps will be repeated 12 times.

Lastly, since the output is still a hidden state a linear layer is applied as the very last step to convert the hidden state to the output. In contrast to the previous steps, this occurs only once [30][31].

4.6 Bidirectional Neural Networks

Bidirectionality allows a neural network to learn input sequences both forward and backward. In sequence data, we have access to both the past and the future input features thus enabling us to utilize them more effectively. A bidirectional layer works as a wrapper for another layer like the LSTM which in the architecture creates two layers of LSTMs that one of them only looks to the data backward and the other forwards.

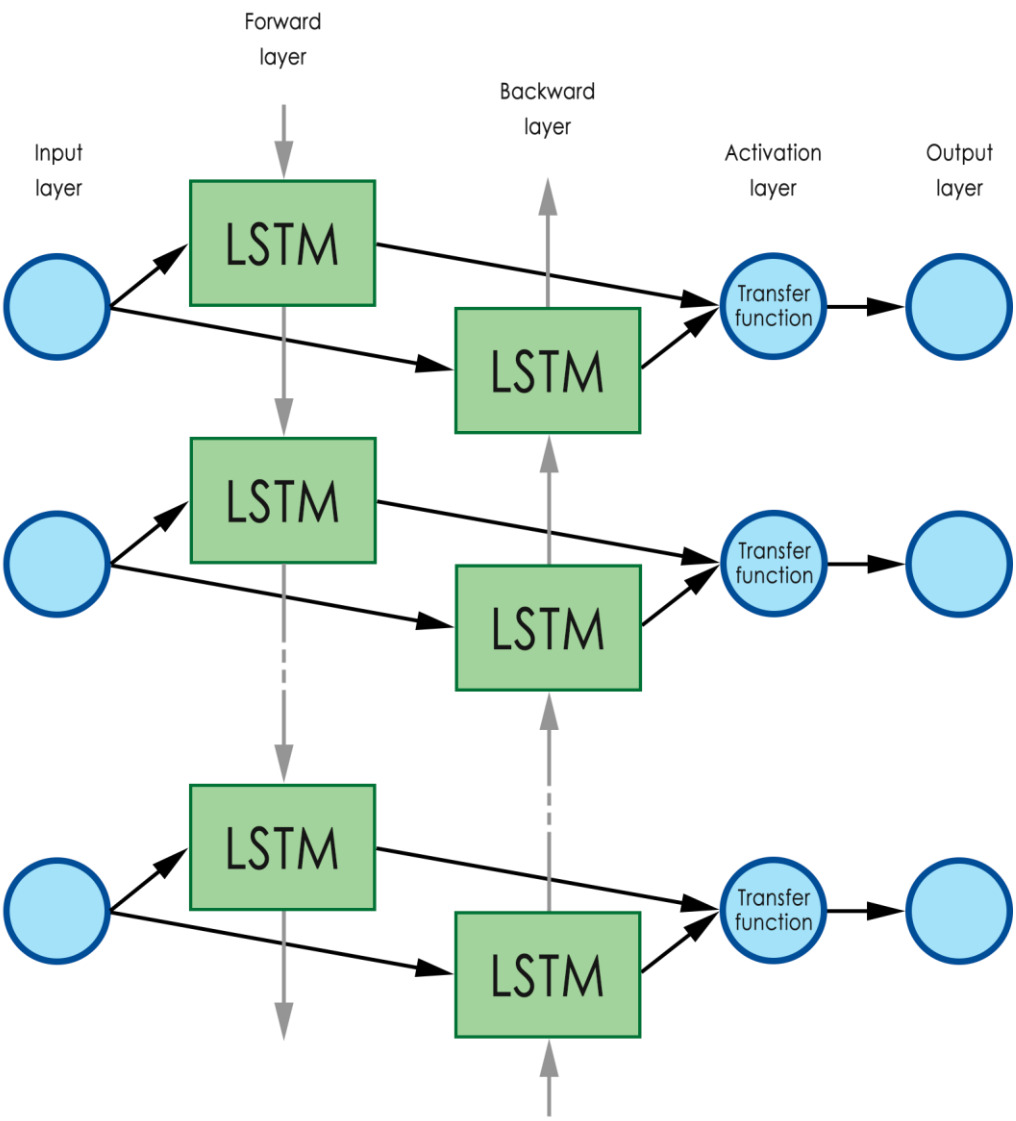


Figure 14. The architecture of the Bidirectional LSTM layer

Both layer outputs are calculated by the LSTM equations and an output vector is generated for every element with the following formula:

$$Y_t = \sigma(\overrightarrow{H_t}, \overleftarrow{H_t})$$

In doing so, we can efficiently make use of past and future features for a specific time frame [32].

5. Air Quality forecasting with Neural Networks

In the current thesis, 6 research papers have been studied where LSTMs were used as part of the model or only to forecast air pollutants. Each paper studies different architectures of Neural Networks and possible methods to improve results.

An LSTM-based aggregated model for air pollution forecasting [33]

Yue-ShanChang et al. gathered information from open databases from the Taiwanese Government and local data from the EPA and CWB stations. They proposed an ALSTM model which uses three layers of inputs and three layers of sub-networks with 128 memory units and returns sequences. The combined output layer optimizes the combined input layer containing the trained local near station data and seemly data to return a single prediction of the $PM_{2.5}$ value of the next 1-8 hour.

They conclude that it is better to predict $PM_{2.5}$ values through their model rather than the standard LSTM, SVR, or GBTR. Still, they believe that the addition of data such as cloud satellite imagery and airflow wind maps will increase the accuracy of their model.

A Bayesian LSTM model to evaluate the effects of air pollution control regulations in Beijing, China [34]

Yang Han et al. collected air quality data in an hourly time range from the database of the US Embassy of Beijing. The proposed machine-learning framework aims to predict the $PM_{2.5}$ concentrations of the city with outputs. One where the country's interventions for pollution are implemented, and a second one where no regulations are implemented. Its architecture consists of an LSTM as its input layer, two fully connected Bayesian layers to map the time trend vector into a vector of continuous values, and a linear layer with a sigmoid function as its output layer.

They noticed that in comparison to non-linear machine learning models their proposed model outperformed them. Still, they suggest that their model can improve by applying a propensity score estimation layer because it is difficult to understand which variables have contributed the most to biases. In addition, they claim that more study is needed to understand the individual effect of a particular regulatory intervention on air quality, and over a particular sector. Lastly, they believe that additional data would be beneficial to improve the accuracy of the model.

A Review of Neural Networks for Air Temperature Forecasting [35]

Trang Thi Kieu Tran et al. reviewed 26 publications related to artificial neural network methodologies to forecast air temperature. They had the following insights. For inputs in the neural network common where temperature, humidity, precipitation, and wind speed as correlated features. The performance of the NN models seemed to be related to the network's configuration, such as hidden neurons and layers. However, they became aware that adding hidden layers and neurons did not always increase accuracy from a number of studies.

Still, they believe that more steps need to be taken to have a clearer image of the matter. For instance, feature selection should be used to decide which meteorological and geographical variables should be given to improve the accuracy of air temperature prediction. They also claim that the combination of neural networks with optimization algorithms could strengthen the model's robustness.

Forecasting PM2.5 Concentration Using a Single-Dense Layer BiLSTM Method [36]

Yeong Min Jang et al. developed an Internet of Things system which collects air data such as temperature, humidity, PM1, and PM2.5 from an industrial-grade sensor which is deployed in a laboratory clean room and then stored in a database. They propose a Single-Dense Layer Bidirectional LSTM which uses two LSTMs that are independent of one another to have both backward and forward information about the sequence at every time step. It uses two hidden layers from two LSTM models to preserve information from both the future and past. It consists of an input layer, a BiLSTM layer and, a single dense layer as its output with a linear activation function to generate prediction with continuous values.

Overall, the proposed model retained great accuracy with various sampling rates. Even though the results were positive they believe further research should be conducted such as taking into account other size particulate matters like $PM_{0.5}$, PM_1, PM_5, and PM_{10}.

A Hybrid Time Series Model based on Dilated Conv1D and LSTM with Applications to PM2.5 Forecasting [37]

Liqing Zhao, Bo Cheng, et al. propose a hybrid model based on LSTM and Dilated Conv1D which can predict future values based on historical data of $PM_{2.5}$ pollutants. Its architecture consists of an input layer that can receive multivariate sequence data, a convolutional layer that consists of one or more layers of dilated convolutional neural in one dimension that can identify the local features of sequences and enhance the dependence between data, an LSTM layer that is fully connected and maps the features into the sample space and an output layer that outputs sequence data of the same latitude as the sample space.

In comparison to standalone networks like LSTM, GRU, and Conv1D the proposed model seems to have the lowest mean square error and the fastest computational performance.

Deep-AIR: A Hybrid CNN-LSTM Framework for Air Quality Modeling in Metropolitan Cities [38]

Yang Han et al. used a variety of different sources as their data. From urban dynamics such as air pollution, meteorology, traffic conditions, and urban morphology to image-like urban dynamics like AQ station maps. They propose a deep learning framework that can estimate up to 24 hours ahead. It consists of a residual CNN which extracts spatial features and their interactions, using a Conv1D that aids the information exchange across different features and an LSTM for creating the time-related dependence of the spatial representations that were extracted for air pollution prediction. They noticed that their proposed model had the best performance in fine-grained air pollution estimation and for air pollution forecasting, however, they noticed that for one-hour predictions the error rate varied between different pollutants.

In the future, they aim to apply the same framework to other locations except for Hong Kong and Beijing to evaluate its performance and its generalization. Furthermore, they wish to conduct further research with other CNN and ConvLSTM models.

6. Data Collection

The reason Europe was selected for forecasting was due to the availability of data and the application of its AQI to multiple countries. If Asia was selected, then each country would have a different AQI and model due to the different inputs and data preparations that it required. In addition, some countries in Asia have jurisdictions that may be subject to copyright, limiting its use or distribution. To begin with, it should be mentioned that all available data that were used were from Open-Source tools and companies which require no license. In addition, they are up to date which enables us to train and test our model with the latest available information.

One of these tools is OpenAQ [39] which enables users to access air quality data from numerous monitors around the world. It is a real-time and historical air quality platform, combining a collection of government-measured and research-grade data which all is entirely open-source. It provides measurements for:

- particulate matter ($PM_{2.5}$)
- particulate matter (PM_{10})
- carbon monoxide (CO)
- nitrogen dioxide (NO_2)
- ozone (O_3)
- sulfur dioxide (SO_2)

They are updated every 10 minutes and, in some locations, even every 2 minutes. It uses data from EEA, EPA, UK-AIR, PurpleAir, AirNow, and many more organizations and universities. A small caveat is that the specific tool allows only 100.000 rows to be given at any call.

Another tool is MeteoStat [40] which is a weather and climate database providing detailed weather data for thousands of weather stations and places around the world. Additionally, archived data is also provided from many old or dismissed weather stations. It provides measurements for:

- The air temperature in °C (temp)
- The dew point in °C (dwpt)
- The relative humidity in percent (rhum)
- The one-hour precipitation total in mm (prcp)
- The snow depth in mm (snow)
- The wind direction in degrees (wdir)

- The average wind speed in km/h (wspd)
- The peak wind gust in km/h (wpgt)
- The sea-level air pressure in hPa (pres)
- The one-hour sunshine total in minutes (tsun)
- The weather condition code (coco)

They are updated hourly and some datasets date back even to the late 18th century. It uses weather and climate data that are provided by the following organizations and systems: Deutscher Wetterdienst, NOAA - National Weather Service NOAA - Global Historical Climatology Network, NOAA - Integrated Surface Database, Government of Canada - Open Data, MET Norway, European Data Portal, and Offene Daten Österreich The data is gathered by different members of the World Meteorological Organization (WMO) and is being shared under the terms of WMO resolution 40 [41].

In this thesis, 4 locations were picked from Europe in order to test the model's performance and generalization. They were selected based on the amount of data that was available from the APIs, the similarity of the geography between them, and the fact that they are some of the most polluted countries on the continent. They were:

- Greece – Athens
- Italy – Naples
- Poland – Gdańsk
- United Kingdom – London

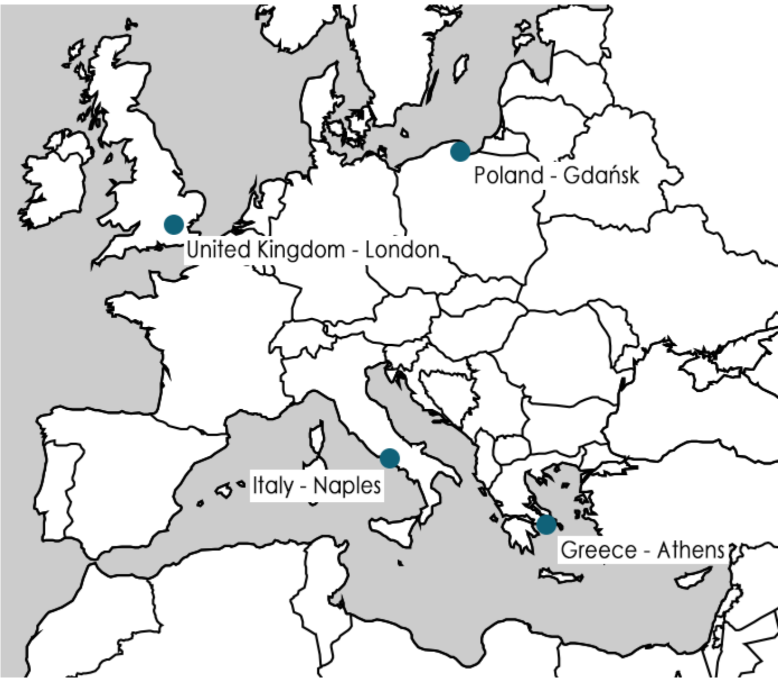


Figure 15. Map of selected locations for data collection

7. Design and development of the LSTM-Conv1D Neural Network

7.1 Introduction

In this chapter, we study methodologies for the design, development, and creation of the Neural Network (NN). At first, we showcase the origins of the Python language, its functionalities, and then the Jupyter Notebook. Furthermore, we showcase how the data was collected and preprocessed in order to be prepared for the model. Subsequently, we give a brief explanation of the libraries that were used for the construction of the NN and data preprocessing.

Afterward, we create different models and determine the methodologies and algorithms that will be used to train them successfully. At the same time, we try different configurations and techniques to produce the best possible result. We compare the results for each variation of the model to find the best optimal structure. It should be noted that every algorithm or methodology that we are going to use will have each own explanation alongside.

Lastly, we use different methodologies to evaluate and measure the performance of the model.

7.2 Python Programming language

Python is an object-oriented, widely-used with dynamic semantics, and high-level programming language, used for AI and general-purpose programming. It was conceived around the late 1980s by Guido van Rossum and its purpose was to be a successor to the ABC programming language.

The very first version of Python was published in February 1991 which apart from exception handling and functions included the core data types of lists, dictionaries, string, and others. It had a module-system and was also object-oriented. Functional programming tools like lambda, map, filter, and reduce were added with Python version 1.0 in January 1994. List comprehensions, a full garbage collector, and Unicode support were included in October 2000, when Python 2.0 was introduced. Python 3.0 came in 2008 as a major release which was not backward compatible with Python 2.x and emphasized the removal of duplicate programming constructs and modules. This action split the community in half. One side of the people was using only Python 2.x religiously and the other was embracing the new Python 3.x and the mindset that it followed. In the year 2020 Python 2.7 stopped getting security patches or other improvements and only Python 3.6.x and later are supported. Python 3.9 is the newest release of the Python programming language that was released in 2021with numerous add-ons such as Module State Access from C Extension Methods, Union Operators in dictionaries, Type Hinting Generics in Standard Collections, Flexible function and variable annotations, Relaxing Grammar Restrictions on Decorators, with improved various features compared to previous versions.

Python is open source and is and is managed by the non-profit organization Python Software Foundation. The language’s name is unrelated to the animal, but an actuality is a reference to an old BBC television comedy sketch series called Monty Python’s Flying Circus. Its goal is to have code that is as understandable as plain English [42].

The language’s design and core philosophy are summarized in the document from the official website of Python called The Zen of Python [43] which includes the following:

- Beautiful is better than ugly.
- Explicit is better than implicit.
- Simple is better than complex.
- Complex is better than complicated.
- Flat is better than nested.
- Sparse is better than dense.
- Readability counts.
- Special cases aren't special enough to break the rules.
- Although practicality beats purity.
- Errors should never pass silently.
- Unless explicitly silenced.
- In the face of ambiguity, refuse the temptation to guess.
- There should be one-- and preferably only one --obvious way to do it.
- Although that way may not be obvious at first unless you're Dutch.
- Now is better than never.
- Although never is often better than *right* now.
- If the implementation is hard to explain, it's a bad idea.
- If the implementation is easy to explain, it may be a good idea.
- Namespaces are one honking great idea -- let's do more of those!

To write the code we used Jupyter Notebooks from a suite of tools named Anaconda. It's an online application that's open-source which allows us to make documents with narrative text, equations, visualizations, and live code. It is used for processing data, visualization, machine learning, and much more [44].

7.3 Use of libraries

The main feature of the Python programming language is developing libraries for any kind of developer work. More specifically, the libraries of Python contain collections of functions and methods that allow many operations to be performed without writing code. It consists of 200+ embedded libraries as well as innumerable libraries which have been created by external users. Below we mention the libraries we used for data analysis and construction of the Neural Network.

- **Numpy** is a Python library specifically designed for scientific computing. It provides a multidimensional array object, various derived objects, and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, and much more. Numpy's optimized and pre-compiled C code that does all the heavy lifting is much faster than regular Python arrays [45].
- **Pandas** is a Python package for data analysis/manipulation that provides data structures designed to make working with both relational and labeled data effectively. It aims to be the top-level component for doing practical and productive, data analysis in Python. It is well suited for tabular data like SQL tables or Excel spreadsheets, ordered and unordered time series data, arbitrary matrix data, and more [46].
- **Matplotlib** is a graphical plotting library for Python which is primarily used for data visualization and its numerical extension NumPy. It is used to create high-quality graphs, charts, and figures that can be static, animated, and interactive. It is a viable alternative to MATLAB, and it allows developers to use its APIs (Application Programming Interfaces) to embed plots in GUI applications [47].
- **Seaborn** is a library used for making statistical graphic representations in Python. It is based on top of Matplotlib and integrates with Pandas data structures. Its main focus is to aid in the exploration and comprehension of data by performing the necessary semantic mapping and statistical aggregations to produce informative and comprehensible plots [48].

- **Scikit-learn (Sklearn)** is a Python library that provides unsupervised and supervised machine learning algorithms like Regression, Classification, Clustering, Model selection, and Preprocessing. It's built upon SciPy, NumPy, Pandas, and Matplotlib. Its goal is to provide a level of robustness and support required for use in production systems and tackle concerns such as ease of use, code quality, collaboration, documentation, and performance [49].
- **Datetime** is a Python module that provides classes and functions that enable the manipulation of dates and times while providing attribute extraction for output manipulation and formatting [50].
- **Time** is a Python module that provides various time-related functions [51].
- **Py-openaq** is a Python wrapper for the OpenAQ API. It enables users to use the API with Python and create objects and arrays containing numerous values such as coordinates, air pollutants, and date records of when they were detected from all over the world. [52]
- **Meteostat** is a Python library that provides access to open weather and climate data using Pandas. It consists of data provided by different public interfaces, most of which are governmental like the National Oceanic and Atmospheric Administration (NOAA), Germany's national meteorological service (DWD), and more. Its data interface provides access to full data dumps (historical observations and statistics) of individual weather stations [53].
- **TensorFlow** is an open-source library for large-scale machine learning, numerical computations, and artificial Intelligence. It provides an abundance of machine and deep learning (aka neural networking) models and algorithms that are available through its API. It provides an API through Python for building applications while executing them in high-performance C++. It uses structures named dataflow graphs which describe how data moves through a series/graph of processing nodes. Each node represents a mathematical operation, and each connection or edge between nodes is a multidimensional data array or tensor [54].

7.4 Data Pre-processing

There are many steps that must be taken for the data before the preparation of the model.

The first step is to select the location of where the data should be collected from. Both tools provide a center point and a radius of how far they should look for monitors that gather data. To avoid getting values from different locations we find the available coordinates from the OpenAQ API and from it select the same location for the MeteoStat.

The second step is to give the time range for both tools. To have a general idea of the air quality the range should be at least a yearlong. The mean value is taken from all the monitors in an hourly step since MeteoStat provides data only hourly. Then both datasets are inspected for gaps in the time range. If any exists, the rolling methodology is applied where they are filled by the mean value of 4 days back of the current step to fill the empty value. After that, the datasets from OpenAQ and MeteoStat are merged according to the index corresponding to the date.

PM_{25}	PM_{10}	NO_2	O_3	SO_2	CO	temp	dwpt	rhum	prcp
22	44.1	35.44	43.16	5.33	450	16.8	12.1	74.1	0.0
…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…
wdir	**wspd**	**pres**	**coco**						
290.0	1.1	1009.6	3.0						
…	…	…	…						
…	…	…	…						

Table 8. The initial feature table of data

The third step is that the AQI is calculated in accordance to the European Air Quality Index for each pollutant and the final AQI representing the state of the air quality for each hour. To make things understandable for the model a key will be given for each category.

A table showing the keys for each classification of the AQI:

Key	Category	$PM_{2,5}$ μg/m³	PM_{10} μg/m³	NO_2 ppb	O_3 ppb	SO_2 ppb
1	Good	0-10	0-20	0-40	0-50	0-100
2	Fair	10-20	20-40	40-90	50-100	100-200
3	Moderate	20-25	40-50	90-120	100-130	200-350
4	Poor	25-50	50-100	120-230	130-240	350-500
5	Very Poor	50-75	100-150	230-340	240-380	500-750
6	Ext. Poor	75-800	150-1200	340-1000	380-800	750-1250

Table 9. The European AQI Table for classification of the AQI value

A table showing the AQI indexes with the other data:

PM_{25}	PM_{10}	NO_2	O_3	SO_2	CO	temp	dwpt	rhum	prcp
22	44.1	35.44	43.16	5.33	450	16.8	12.1	74.1	0.0
…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…
wdir	**wspd**	**pres**	**coco**	**PM_{25} _AQI**	**PM_{10} AQI**	**NO_2 AQI**	**O_3 AQI**	**SO_2 AQI**	**Overall AQI**
290.0	1.1	1009.6	3.0	3	3	1	1	1	3
…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…

Table 10. The feature table of data with AQIs of each pollutant

The next step is the use of Data Correlation is applied using the following methods: Pearson, Spearman, Kendall

Pearson's Correlation Coefficient [55] is also known as the Pearson Product-Moment Correlation Coefficient. It is a way of measurement of the linear relationship between two random variables - X and Y. From a mathematical perspective, if (σ_X) is considered as the standard deviation of X and, (σ_{XY}) as the covariance between X and Y, then the Pearson's correlation coefficient ρ is given by:

$$\rho_{X,Y} = \frac{\sigma_{X,Y}}{\sigma_X \sigma_Y}$$

Since the covariance is always smaller than the product of the individual standard deviations, the value of ρ fluctuates between -1 and +1. From the above it can be determined that the correlation of a variable with itself is:

$$\rho_{X,X} = \frac{\sigma_{X,Y}}{\sigma_X \sigma_X} = 1$$

Spearman's rank Correlation Coefficient [56] quantifies the degree to which ranked variables are associated by a monotonic function, meaning an increasing or decreasing relationship. As a statistical hypothesis test, it assumes that the samples are uncorrelated with one another.

The intuition is that it calculates a Pearson's correlation by using the rank values contrary to the real values. The Pearson's correlation is the calculation of the covariance between the two variables that are normalized by the variance or spread of both.

There are two methods to calculate Spearman's correlation depending on whether our data does not have tied ranks, or our data has tied ranks. In the case where there are no tied ranks, the following formula is used:

$$\rho = 1 - \frac{6\Sigma d_i^2}{n(n^2-1)}$$

Where n = number of cases and d_i = difference in paired ranks. The formula to use for tied ranks is:

$$\rho = \frac{\Sigma_i(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\Sigma_i(x_i-\bar{x})^2\Sigma_i(y_i-\bar{y})^2}}$$

where i = paired score.

Kendall's Correlation Coefficient [57] is used to assess the similarities between the ordering of data in accordance with the ranking of quantities. In comparison to other types of correlation coefficients where they calculate their correlation using the observations as the basis, Kendall's correlation uses pairs of observations and ascertains the strength of association in accordance with the pattern of concordance and discordance between the pairs.

- Concordant: Ordered in the same way (consistency). If (x2 — x1) and (y2 — y1) have the same sign, then the pair of observations is considered concordant.
- Discordant: Ordered differently (inconsistency). If (y2 — y1) and (x2 — x1) have opposite signs, then the pair of observations are treated as concordant.

It is given by:

$$T_B = \frac{n_c - n_d}{\sqrt{(n_0-n_1)(n_0-n_2)}}$$

n_c = Number of concordant pairs,

n_d = Number of discordant pairs,

$n_0 = n(n-1)/2,$

$n_1 = \sum_i t_i(t_i-1)/2,$

$n_2 = \sum_j u_j(u_j-1)/2$

As a result, the main 5 pollutants are compared with each value from OpenAQ, MeteoStat and if there is no positive or negative correlation they are removed. This is done to avoid computations that have little to no effect on the final product. The current values that will be taken into consideration for the training of the Neural Network:

PM_{25}	PM_{10}	NO_2	O_3	SO_2	CO	temp	dwpt	rhum	prcp
22	44.1	35.44	43.16	5.33	450	16.8	12.1	74.1	0.0
…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…
wdir	**wspd**	**pres**	**coco**	**PM_{25} _AQI**	**PM_{10} AQI**	**NO_2 AQI**	**O_3 AQI**	**SO_2 AQI**	**Overall AQI**
290.0	1.1	1009.6	3.0	3	3	1	1	1	3
…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…

Table 11. The current features of the dataset

Before the data is passed to the Neural Network model it needs to go through specific procedures to be able to go through the model.

First, the data needs to be split to features and targets to select the values that need to be predicted. With d_length the length of the dataset, n_features the number of available features, and n_selected_features the number of selected features for prediction the following shapes are:

Features = (d_length, n_features)

Targets = (d_length, n_selected_features)

Then with the tool named train_test_split, the data are split up into training (80%), testing (10%), and validation data (10%) to assess the loss and accuracy of the model.

X_train = (d_length * 80%, n_features)

X_test = (d_length * 10%, n_features)

X_val = (d_length * 10%, n_features)

y_train = (d_length, n_selected_features)

y_test = (d_length * 10%, n_selected_features)

y_val = (d_length * 10%, n_selected_features)

Then the data needs to be normalized to improve the stability, training process, and accuracy of the model. Lastly, the time series data needs to be re-framed as a supervised learning problem by using the sliding window [58] method with the TensorFlow's TimeseriesGenerator so that the model can learn the patterns more accurately.

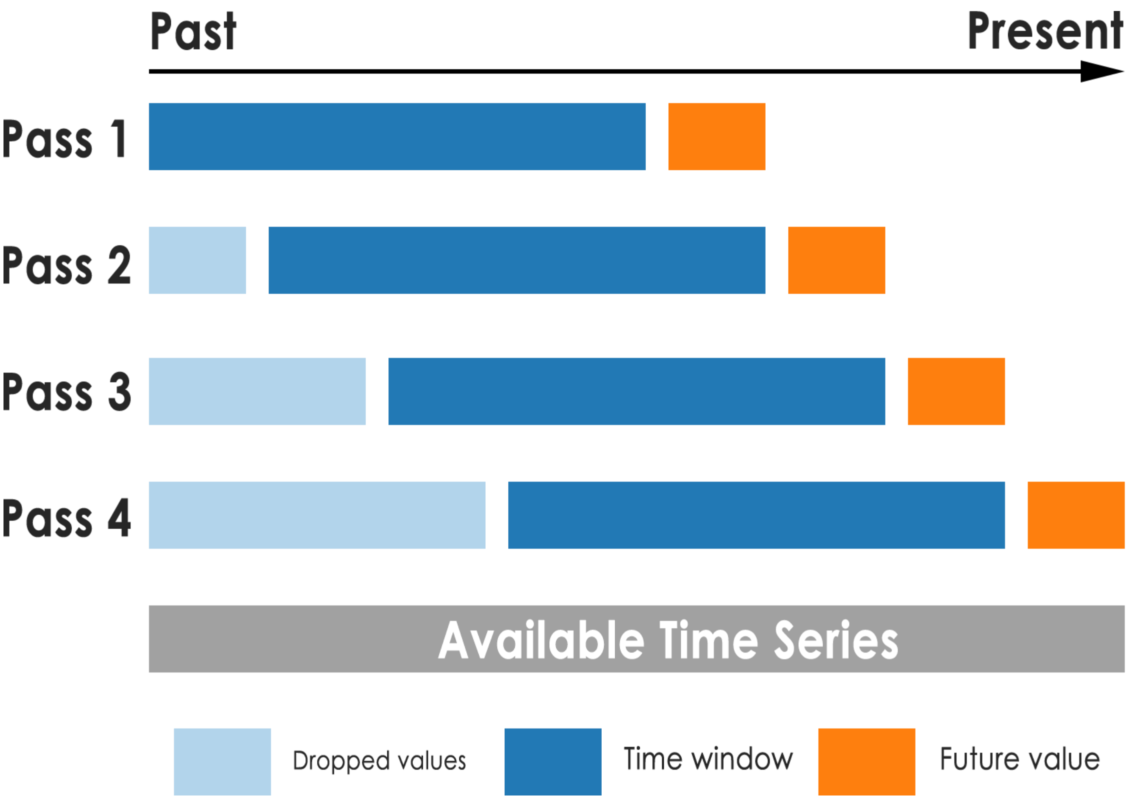


Figure 16. Sliding window on a time series

The final shape of the train, validation, and test datasets is (batch size, window size, n_features).

7.5 Neural Network conception

The goal of the Neural Network model is to be able to accept a number of features from the OpenAQ and MeteoStat datasets and be able to predict the values of PM25, PM10, NO2, O3, SO2. In other words, the model needs to take multivariate series data and output multiple values. To achieve that, The structure of the final Neural Networks went through numerous variations and testing scenarios before reaching its final architecture. Inspired from the papers mentioned in chapter 8 the parts that were used were the following:

- Bidirectional wrapper [59] in order to take advantage of the time-series data that will enter the model.
- LSTM layer to capture the trends and seasonality of the data.
- Dropout layer [60] to regularize the neural network.
- Conv1D [61] layer to capture the patterns and spikes of the data.
- LeakyReLU [62] layer as an activation function to avoid sparse gradients.
- MaxPooling1D [63] layer to calculate the largest value in each patch of each feature map.
- Flatten [64] layer to reshape the tensor to have a shape that is equal to the number of elements.
- Dense layer for the outputs of the model.
- ReduceLROnPlateau [65] callback to monitor the learning rate of the model and if no improvement is seen then the learning rate is reduced.
- EarlyStopping [66] callback to stop training when the model stopped improving.

Afterwards the following architectures were created:

a) Conv1D – Conv1D

b) BiLSTM – BiLSTM

c) BiLSTM – BiLSTM – Conv1D – Conv1D

d) BiLSTM – BiLSTM – Conv1D

e) BiLSTM – Conv1D

f) BiLSTM – Conv1D – Conv1D

g) Conv1D – BiLSTM

h) Conv1D – Conv1D – BiLSTM

i) Conv1D – Conv1D – BiLSTM – BiLSTM

j) Conv1D – BiLSTM – BiLSTM

For each the common attributes were the optimizer (Adam), the number of neurons/filters for each layer (200 for the LSTM and 128 for the Conv1D), and the window length of visible data per prediction (12).

Below is a chart showcasing each model performance:

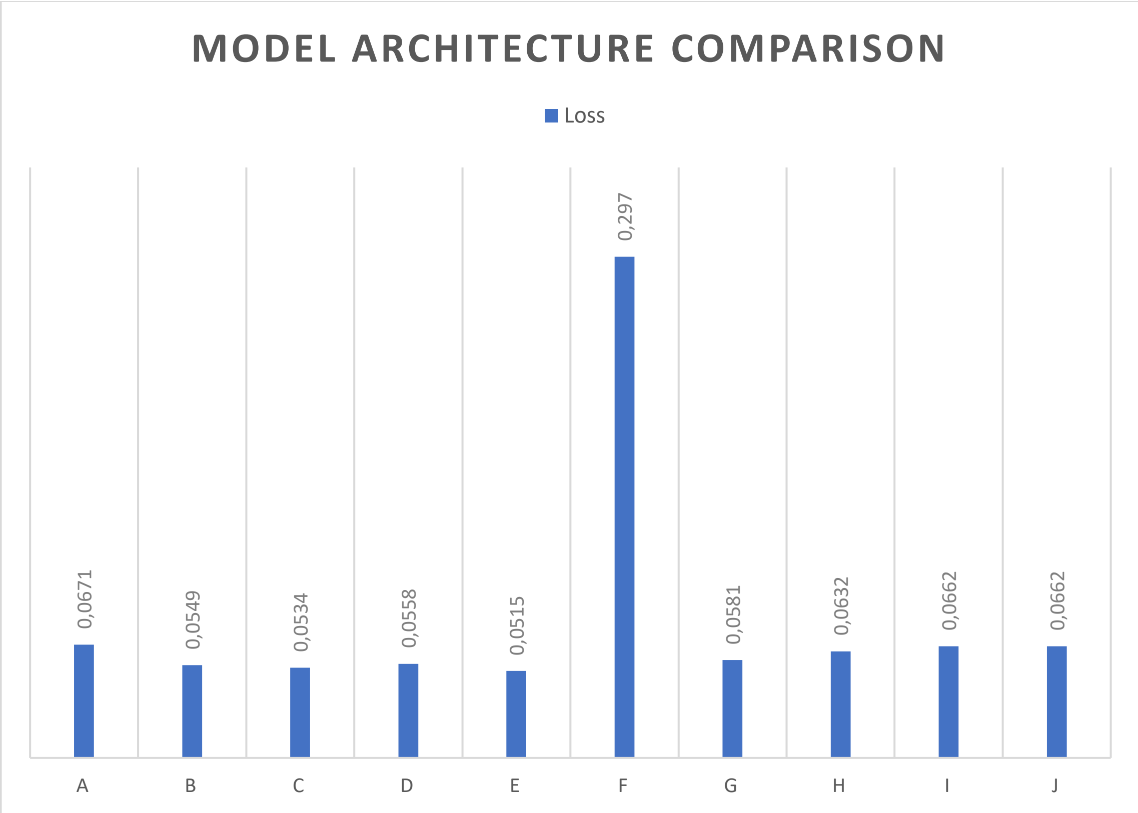


Figure 17. Chart showing performance comparison of model architectures

It seems that model (E) had the lowest loss value. Still, for the next steps of experimentation, the (A) and (B) models will be used as a comparison to the performance of the proposed model. The next step was to compare optimizers for the models. The ones that were selected were:

- Adam [67] is a stochastic gradient descent method that uses an adaptive estimation of first and second-order moments. It is considered a good choice to begin experimenting with since it can handle large data and parameters.
- RMSprop [68] is an optimizer that keeps a moving average of the square of gradients and divides it by the root of that average. It is considered a more efficient choice for embedding lookup tables that are large.
- Adagrad [69] is an optimizer that during training adapts its learning rate relative to how often a parameter is updated.
- Adadelta [70] is a stochastic descent method that uses an adaptive learning rate to avoid learning decay throughout training and the avoidance of manual global learning rate selection. It is considered an extension of Adagrad that uses the moving window of the gradient to adapt its learning rate.

- Adamax [71] is a variant of the Adam optimizer and is considered superior when it comes to embedding models.
- Nadam [72] is an optimizer similar to Adam but uses the Nesterov momentum.
- FTRL [73] is an optimizer most suitable for models with features spaces that are large and sparse.

Below is a chart showcasing the loss for each optimizer:

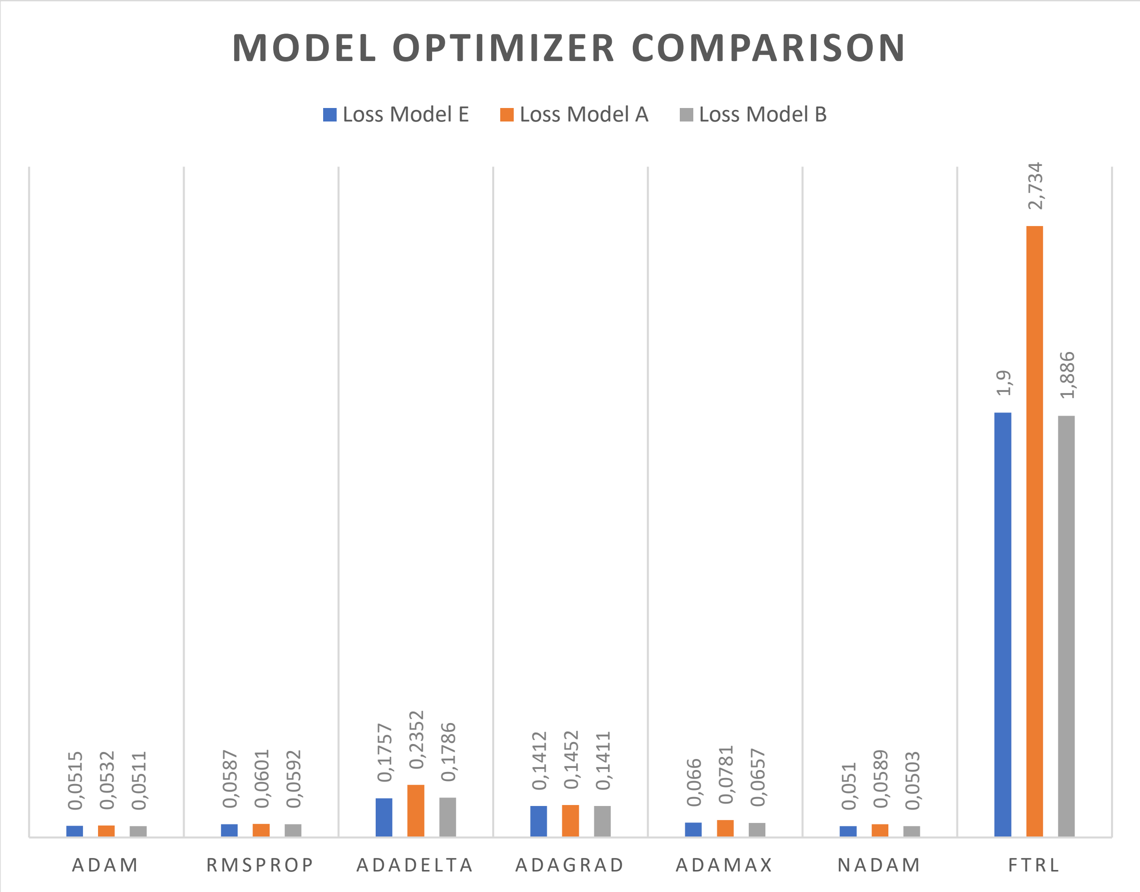


Figure 18. Comparison of optimizers for the main models

It seems the Nadam optimizer seems the best option to increase the accuracy of the model.

Furthermore, the inputs from the data should be compared to find the best features for the models. The use cases took into consideration the following scenarios:

- No AQI data – The model will be trained with all data except the AQI indexes for each pollutant
- No weather data – The model will be trained with all data except the weather data from MeteoStat

- No AQI + weather data – The model will be trained without the above mentioned previously.
- temp – The model will be trained with the main pollutants and the temperature feature only
- dwpt – The model will be trained with the main pollutants and the dew point feature only
- rhum – The model will be trained with the main pollutants and the relative humidity feature only
- prcp – The model will be trained with the main pollutants and the one-hour total precipitation feature only
- wdir – The model will be trained with the main pollutants and the wind direction feature only
- wspd – The model will be trained with the main pollutants and the wind speed feature only
- pres – The model will be trained with the main pollutants and the sea-level air pressure feature only
- coco – The model will be trained with the main pollutants and the weather condition code feature only

Below is a chart showcasing the result of each experiment with the 3 selected models:

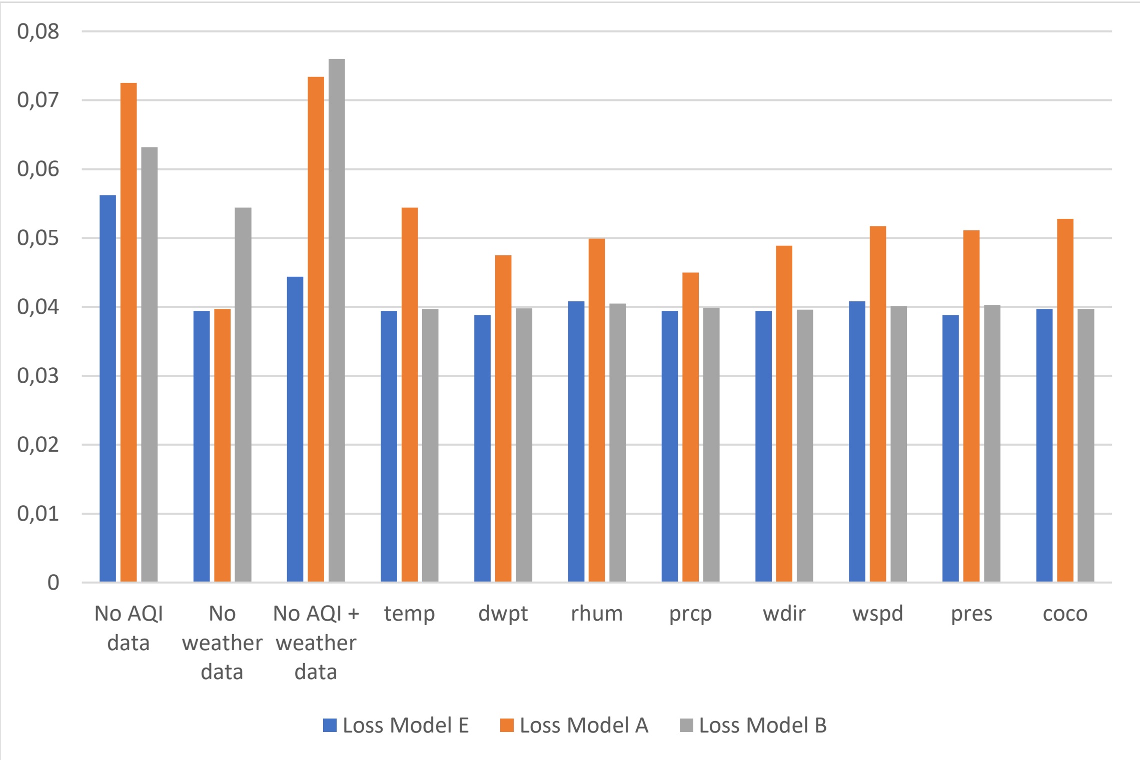


Figure 19. Comparison of performance of input features

Consequently, the chart and the previous correlation coefficient measurements can help us conclude that the best features that have the best correlation are:

- AQI indexes,
- temp,
- rhum,
- wspd
- pres
- coco

The next step was to find out what the best time window would be to capture the pattern, trend, and seasonality of the time series in accordance to the previously mentioned selected features.

Through the TimeseriesGenerator the window size is adjusted for the training, test, validation dataset to create the equivalent shapes.

Therefore, the following assessment took place where the models were trained with a window size of 5 to 50 hours from the past to make predictions. Below is a line chart that shows how the loss comparison followed:

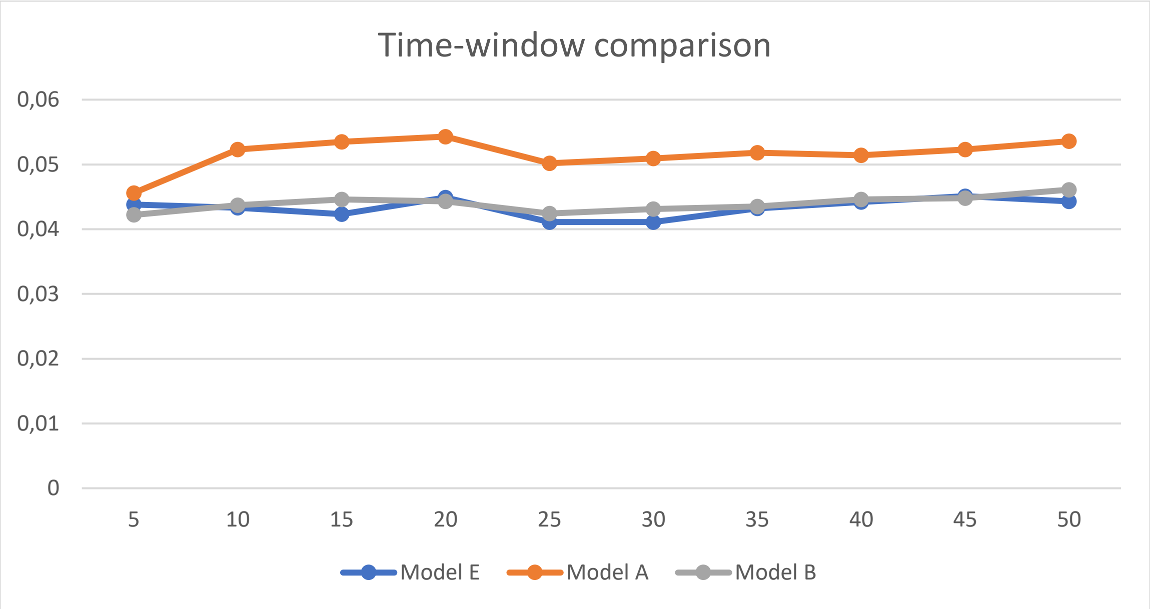


Figure 20. Comparison of different time-windows on the model

It seems that the best time windows range from 10-15 hours and 25-30 hours to make a better prediction. For our testing purposes, we chose to have a 12-hour time window.

The next step was to test the generalization of the model by using it in other locations as mentioned previously in chapter 5.

Below is a chart showcasing the value for each country:

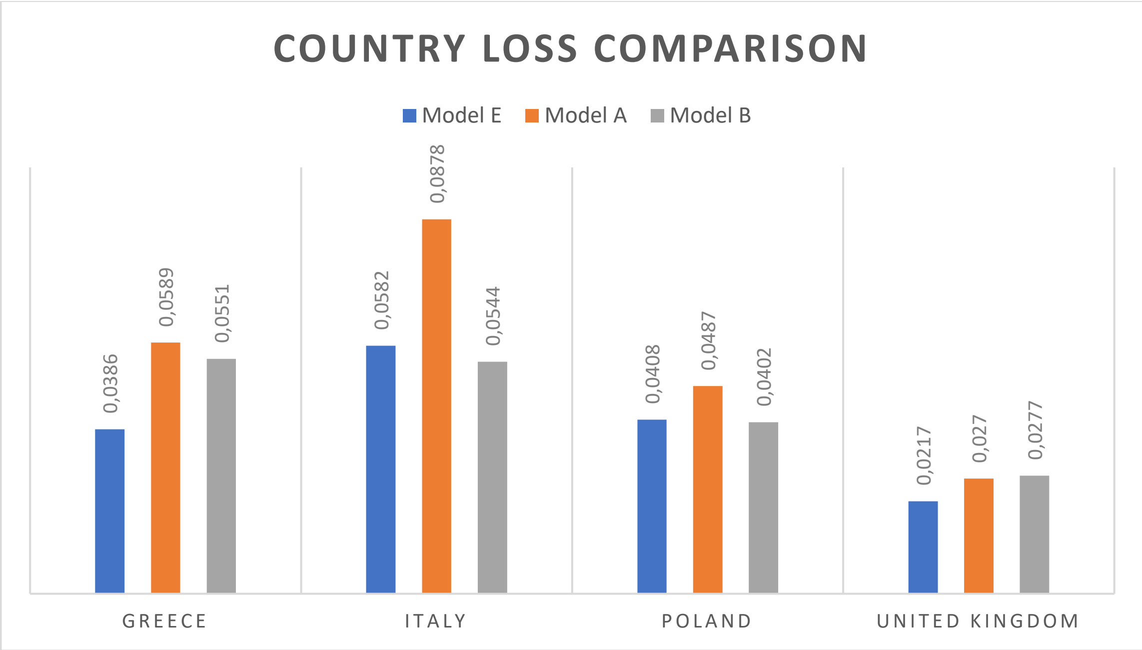
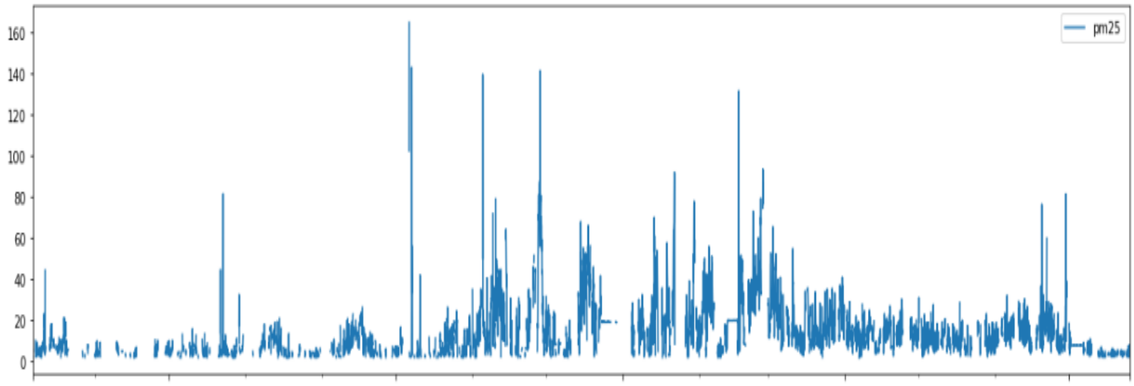
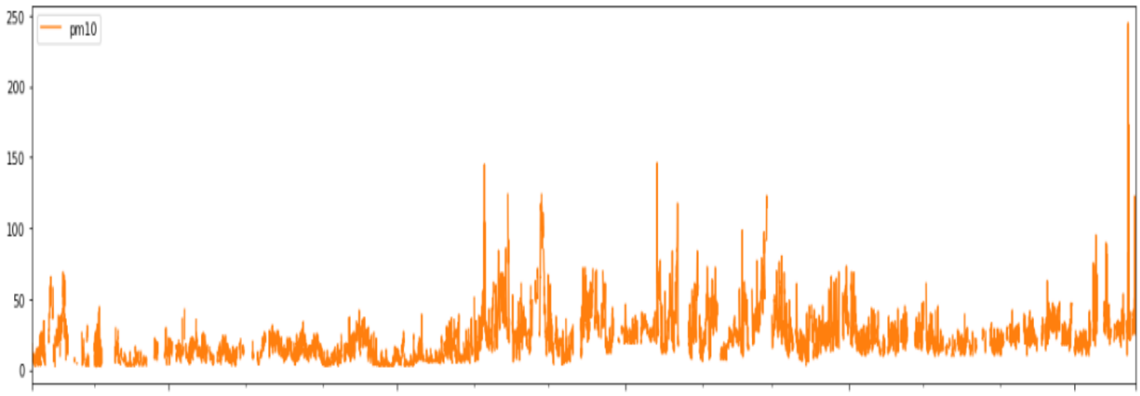


Figure 21. Comparison of model's performance on different countries

The model is able to generalize very well in other locations. The only issue that seems to affect its accuracy is the missing values from each location's dataset. For instance, the data of Italy seem to have a lot of holes between sequences.





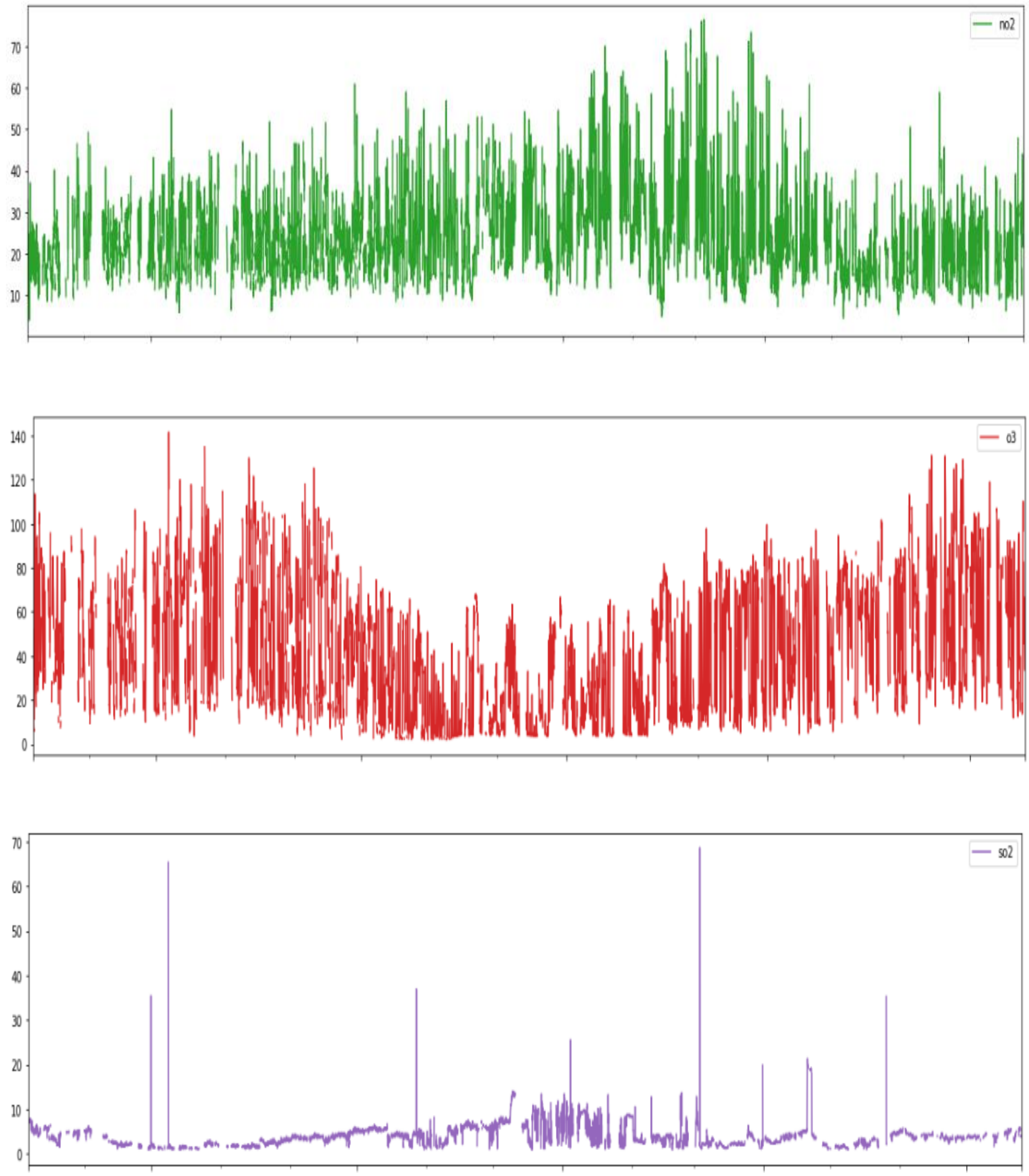


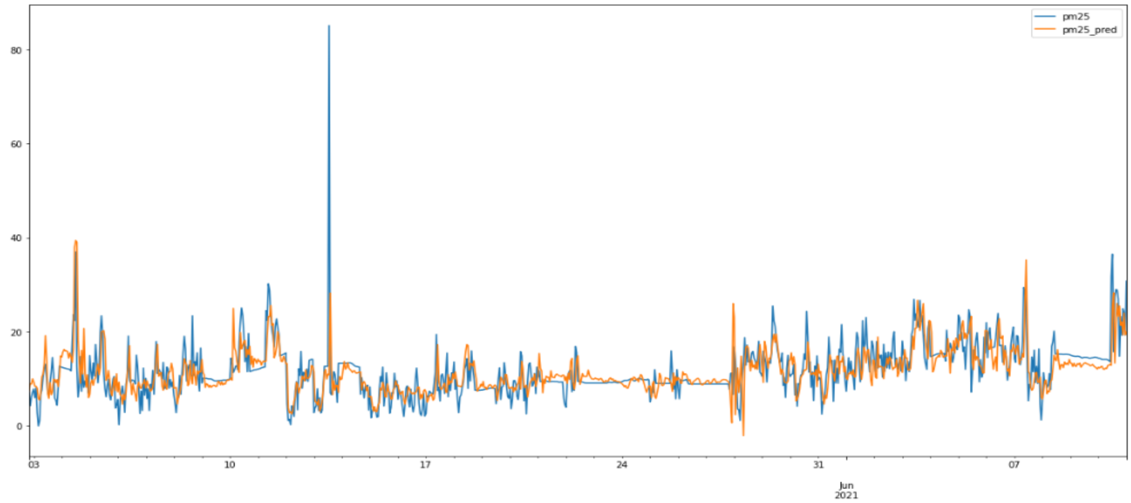
Figure 22. Charts showing missing inputs of data in Italy

Sadly, that is something we must fill by hand, and it is notable that it affects the performance of the model.

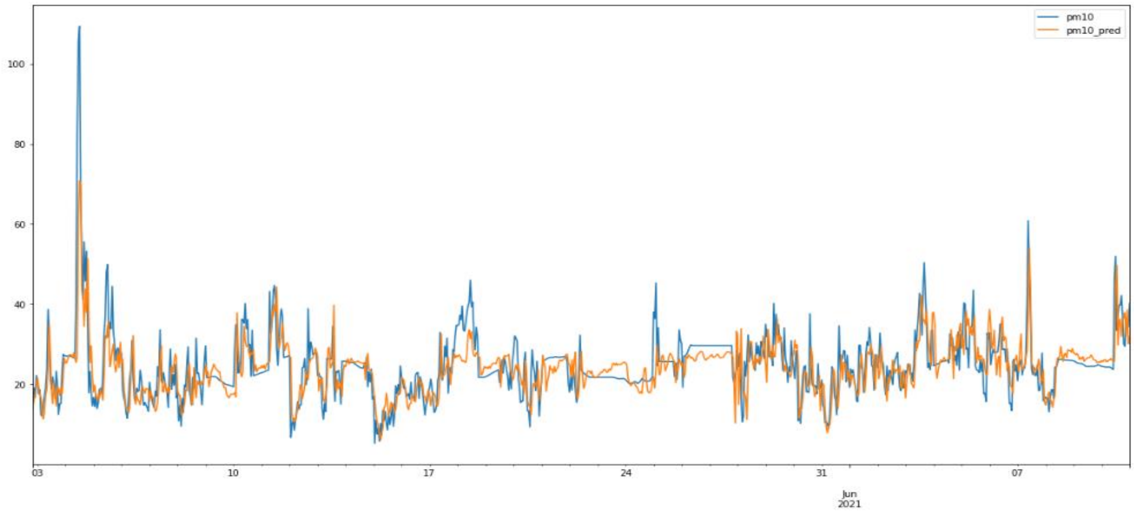
Throughout Europe, the countries that were selected seemed the ones with the least missing data in locations similar to one another. The same issue can be seen in the data used for making predictions.

For example, below are with orange color the predicted values of the model and with blue the actual values of the pollutants in Italy:

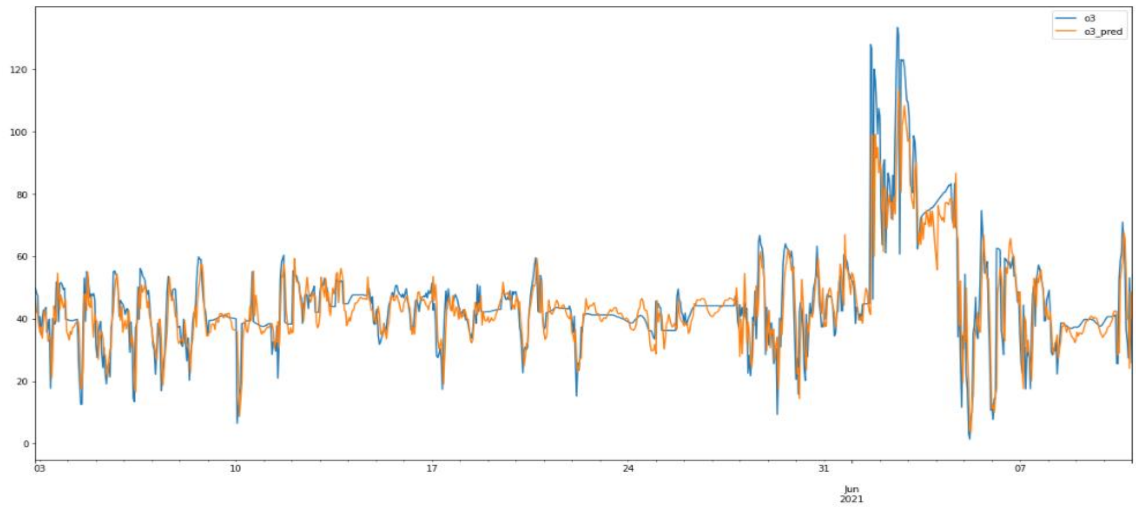
- $PM_{2.5}$



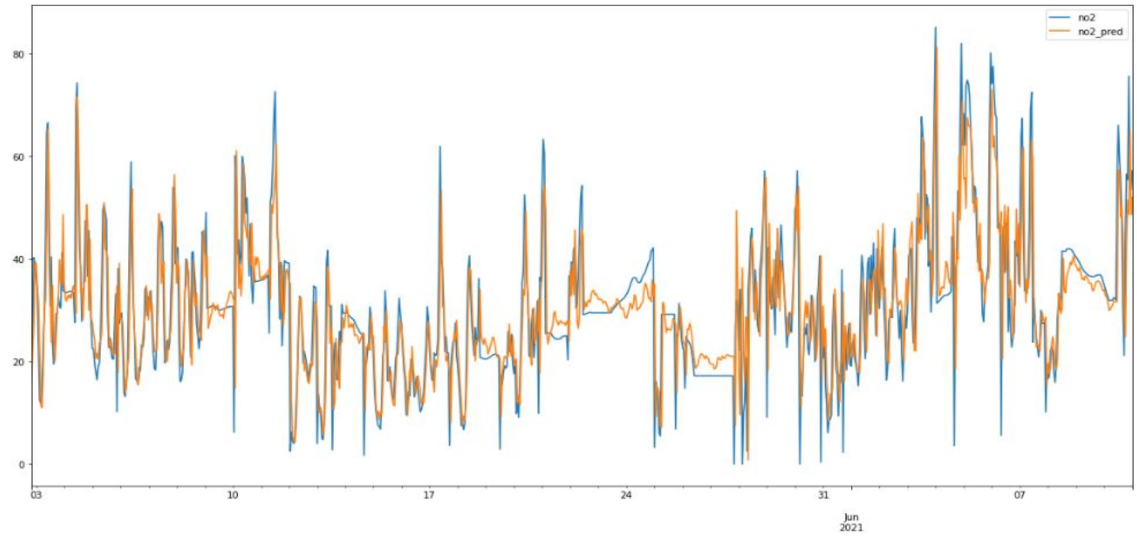
- PM_{10}



- No_2



- O_3



- SO_2

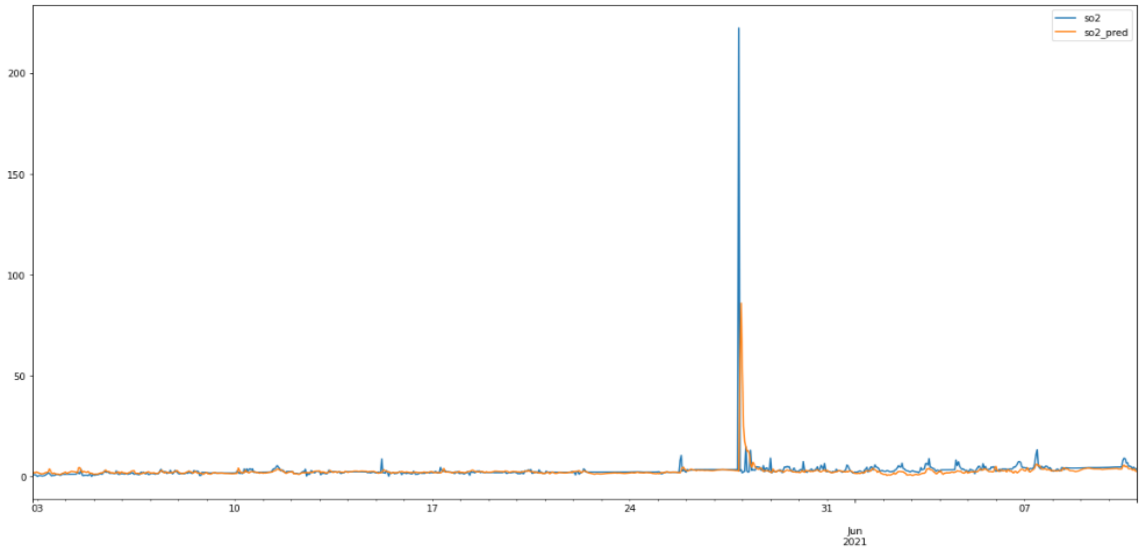


Figure 23. Charts showing the performance of the model on testing data

For extreme situations where the values have sudden spikes, the model is able to detect when that happens pretty accurately and at parts of the data where they were missing values and were filled by hand, the model still generalizes what behavior it should have followed.

It should be noticed that the model has greater performance when it is given a greater amount of data.

As of now, all tests have been executed with a time range of 13 months (10/5/2020 – 10/6/2021). When the amount of data was increased to 16 months (10/5/2020 – 10/9/2021) there was an increase in performance throughout all the countries.

Below is a table showing the difference between the different time ranges:

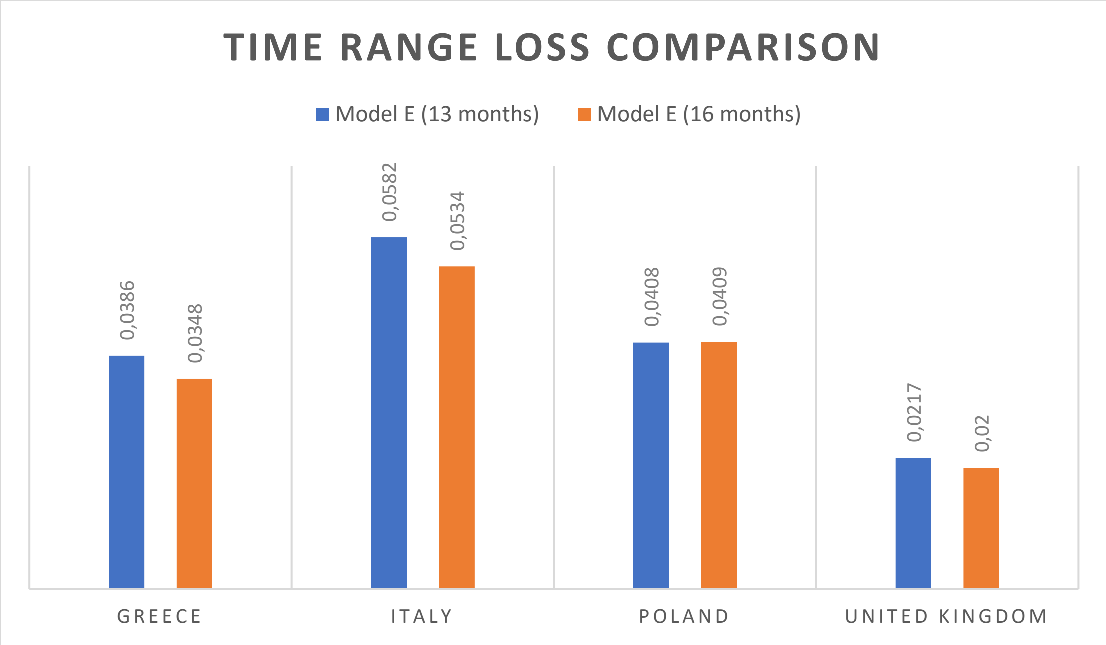


Figure 24. Performance comparison of the model with different time ranges of data

We believe that the model can increase its performance if even more data become available.

7.6 Neural Network evaluation

The last thing that remains is to evaluate the model's prediction with various metrics of error.

More specifically, the following formulas and measurements were applied for each country to calculate how accurate the model could predict each pollutant:

- mean absolute error (MAE) [74] calculates the mean of the absolute difference between labels and predictions.
- mean squared error (MSE) [75] calculates the mean of square errors between labels and predictions and then the last dimension is returned.
- root mean squared error (RMSE) [74] calculates the root mean squared error between true and predicted values.
- mean absolute percentage error (MAPE) [75] calculates the mean absolute error between true and predicted values.
- mean squared logarithmic error (MSLE) [75] calculates the mean squared logarithmic error between the true and predicted values.

Below table showcasing the error for each pollutant:

	$PM_{2.5}$	PM_{10}	NO_2	O_3	SO_2
	GREECE				
MAE	2.3510184	85.21125	9.230994	11.205703	0.03187468
MSE	3.1101303	51.342964	7.1654005	12.285092	0.028357796
RMSE	3.2918816	23.703875	4.8686624	22.65905	0.0657287
MAPE	5.574972	62.167572	7.8846416	7.192999	0.011438048
MSLE	0.92186993	1.8138984	1.3468105	19.495068	0.036511477
	ITALY				
MAE	3.6354566	5.0125604	5.58043	6.5661182	0.7426315
MSE	79.52773	152.14317	62.708908	79.69209	1.0333923
RMSE	8.917832	12.334633	7.9188957	8.927043	1.016559
MAPE	21.01744	14.017476	17.569284	28.939566	17.802923
MSLE	0.06863251	0.039954823	0.051989093	0.07896093	0.037769858
	POLAND				
MAE	1.0016118	2.4516788	2.112318	5.9054646	0.42533332
MSE	2.2924168	14.514561	11.562344	77.549805	0.48306033
RMSE	1.5140729	3.809798	3.4003446	8.806236	0.69502544
MAPE	10.758018	18.438368	17.683176	18.473846	18.283915
MSLE	0.016640462	0.043241985	0.045251716	0.0718942	0.027460048
	UNITED KINGDOM				
MAE	0.9295081	1.9746488	1.9923686	3.8194306	0.4288801
MSE	2.4657893	7.8063893	8.676858	32.139717	3.262888
RMSE	1.5702832	2.7939916	2.9456506	5.66919	1.8063465
MAPE	9.329815	10.249915	9.506956	10.010791	19.589449
MSLE	0.013089749	0.016862972	0.016125409	0.01987558	0.03427104

Table 12. Table showing error ratings for pollutants and countries

It can be confirmed that the model is able to come close to the actual values of the time series with a minimum margin of error.

For Autocorrelation, if we zoom close to the predictions of one of the pollutants like PM_{25} we are able to see that the correlation of the predicted values and the actual values is closer to -1 rather than 1.

Below a graph showing 5 values:

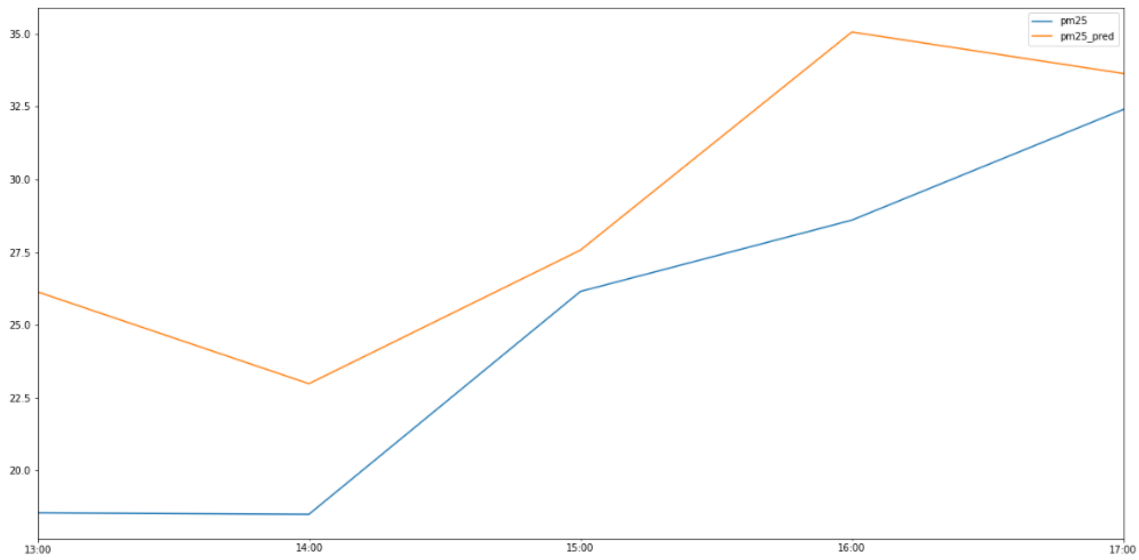


Figure 25. The Autocorrelation of the final model

It is true however that the model's predictions still hold a certain amount of uncertainty on their accuracy.

For Aleatoric uncertainty, big factors that can affect the model are numerous. For instance, forest fires which even though occur mainly in summer still are unpredictable in the volume and duration that they will last. As a result, their impact on the AQI's location can be unpredictable on how long it will hold and how high the pollutant's values will reach.

For Epistemic uncertainty, we are able to showcase that by using different time ranges to the training of the model we can increase its accuracy. Thus, we can claim that this type of uncertainty can be reduced. Not to mention the fact that, the addition of other types of features can reduce it even more.

7.7 Conclusion and Future Work

Having completed the research of Neural Networks utilizing the datasets with air metrics and trying to adjust them to achieve the best possible result, we are called to take advantage of a large number of techniques, hyperparameters, and algorithms. Studying the theory on NNs, for our research we used the techniques and methodologies we consider appropriate, and which extract the best results in a supervised problem. We can conclude with confidence from the rigorous experimentation that the final product is adequate to tackle real life problems and that it can be used in end-to-end applications.

Below an image showing the final architecture of the BiLSTM – Conv1D network:

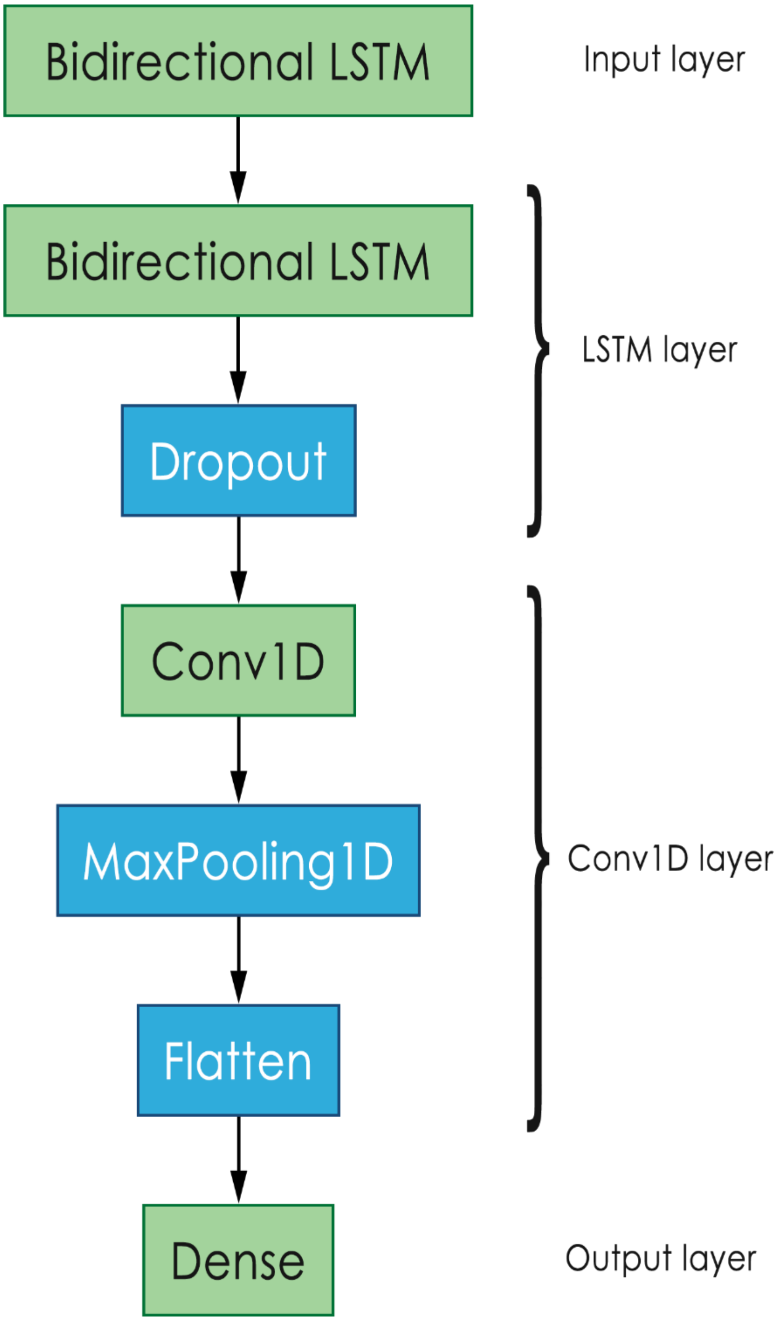


Figure 26. The final architecture of the BiLSTM-Conv1D

Future goals would be to do additional research with other locations, greater quantity of data, and experiment with different time steps and strides on the dataset. We could also use other methodologies like multi-step forecasting to see how far the model can predict the future.

Finally, we could have more entries in the set or have datasets from other global centers to compare existing algorithms in different sets of data.

Source for the code used: https://github.com/GeorgeCodeHub/Analysis-and-prediction-of-air-pollution-using-BiLSTM-Conv1D

8. References

[1] – U.S. National Park, https://www.nps.gov/subjects/air/index.htm, Last Visited: 21/9/2021

[2] – U.S. National Park, *Where Does Air Pollution Come From*, 2018, https://www.nps.gov/subjects/air/sources.htm, Last Visited: 19/9/2021

[3] – NRDC, *Air Pollution: Everything You Need to Know*, 2021, https://www.nrdc.org/stories/air-pollution-everything-you-need-know, Last Visited: 21/9/2021

[4] – European Environment Agency, *National Emission reduction Commitments Directive*, https://www.eea.europa.eu/themes/air/air-pollution-sources-1/national-emission-ceilings , Last Visited: 21/9/2021

[5] – U.S. Environmental Protection Agency, *Air Pollution: Current and Future Challenges*, https://www.epa.gov/clean-air-act-overview/air-pollution-current-and-future-challenges, Last Visited: 21/9/2021

[6] – Climate & Clean Air Pollution, *Air Pollution in Asia and the Pacific: Science-based solutions*, 2019, https://www.ccacoalition.org/en/resources/air-pollution-asia-and-pacific-science-based-solutions-summary-full-report, Last Visited: 21/9/2021

[7] – Felix Leung, *How China is Winning Its Battle Against Air Pollution,* 2021, https://earth.org/how-china-is-winning-its-battle-against-air-pollution/, Last Visited: 21/9/2021

[8] – Wikipedia, *Air quality index*, https://en.wikipedia.org/wiki/Air_quality_index, Last Visited: 21/9/2021

[9] – National Weather Service, https://www.weather.gov/safety/airquality-aqindex, Last Visited: 21/9/2021

[10] – European Environment Agency, https://airindex.eea.europa.eu/Map/AQI/Viewer/, Last Visited: 21/9/2021

[11] – U.S. Environmental Protection Agency, https://www.epa.gov/outdoor-air-quality-data, Last Visited: 21/9/2021

[12] – AirNowtech, https://forum.airnowtech.org/t/aqi-calculations-overview-ozone-pm2-5-and-pm10/168, Last Visited: 21/9/2021

[13] – IQAir, *Can air pollution be predicted*, 2021, https://www.iqair.com/us/blog/air-quality/can-air-pollution-be-predicted, Last Visited: 21/9/2021

[14] – Tableau, *Time Series Forecasting: Definition, Applications, and Examples*, https://www.tableau.com/learn/articles/time-series-forecasting, Last Visited: 21/9/2021

[15] – TensorFlow, *Time series forecasting*, https://www.tensorflow.org/tutorials/structured_data/time_series, Last Visited: 21/9/2021

[16] – Marco Peixeiro, *The Complete Guide to Time Series Analysis and Forecasting*, 2019, https://towardsdatascience.com/the-complete-guide-to-time-series-analysis-and-forecasting-70d476bfe775, Last Visited: 21/9/2021

[17] – Tim Smith, *Autocorrelation*, 2021, https://www.investopedia.com/terms/a/autocorrelation.asp, Last Visited: 2/10/2021

[18] – Issac Faber, *Why you should care about the Nate Silver vs. Nassim Taleb Twitter war*, 2018 https://towardsdatascience.com/why-you-should-care-about-the-nate-silver-vs-nassim-taleb-twitter-war-a581dce1f5fc, Last Visited: 2/10/2021

[19] – Wikipedia, *Uncertainty quantification*, https://en.wikipedia.org/wiki/Uncertainty_quantification#Aleatoric_and_epistemic_uncertainty, Last Visited: 2/10/2021

[20] –IBM, Neural Networks, *Perceptron*, https://www.ibm.com/cloud/learn/neural-networks, Last Visited: 21/9/2021

[21] – Wikipedia, *Perceptron*, https://en.wikipedia.org/wiki/Perceptron, Last Visited: 21/9/2021

[22] – Jason Brownlee, *Crash Course on Multi-Layer Perceptron Neural Networks*, 2020, https://machinelearningmastery.com/neural-networks-crash-course/, Last Visited: 21/9/2021

[23] – Wikpedia, *Multilayer perceptron*, https://en.wikipedia.org/wiki/Multilayer_perceptron, Last Visited: 21/9/2021

[24] – Sumit Saha, *A comprehensive Guide to Convolutional Neural Networks*, 2018 https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53, Last Visited: 21/9/2021

[25] – Shiva Verma, *Understanding 1D and 3D Convolution Neural Network*, 2019, https://towardsdatascience.com/understanding-1d-and-3d-convolution-neural-network-keras-9d8f76e29610, Last Visited: 21/9/2021

[26] – e2eml, *Convolution in one dimension for neural networks, 2019,* https://e2eml.school/convolution_one_d.html, Last Visited: 25/9/2021

[27] – Niklas Donges, *A Guide to RNN: Understanding Recurrent Neural Networks and LSTM Networks*, 2021, https://builtin.com/data-science/recurrent-neural-networks-and-lstm, Last Visited: 21/9/2021

[28] – IBM, *Recurrent Neural Networks*, https://www.ibm.com/cloud/learn/recurrent-neural-networks, Last Visited: 21/9/2021

[29] – Ashray Saini, In-Depth Explanation of Recurrent Neural Network, 2021 https://www.analyticsvidhya.com/blog/2021/07/in-depth-explanation-of-recurrent-neural-network/, Last Visited: 30/9/2021

[30] – Colah, *Understanding LSTM Networks*, 2015, https://colah.github.io/posts/2015-08-Understanding-LSTMs/, Last Visited: 21/9/2021

[31] – Rian Dolphin, *LSTM Networks | A Detailed Explanation*, 2020, https://towardsdatascience.com/lstm-networks-a-detailed-explanation-8fae6aefc7f9, Last Visited: 21/9/2021

[32] – Dive into Deep Learning, https://d2l.ai/chapter_recurrent-modern/bi-rnn.html, Last Visited: 21/9/2021

[33] – Yue-Shan Chang, Hsin-Ta Chiao, Satheesh Abimannan, Yo-Ping Huang, Yi-Ting Tsai, Kuan-Ming Lin, *An LSTM-based aggregated model for air pollution forecasting*, 2020, https://www.sciencedirect.com/science/article/pii/S1309104220301215, Last Visited: 21/9/2021

[34] – Yang Han, Jacqueline CK Lam, Victor OK Li David Reiner, *A Bayesian LSTM model to evaluate the effects of air pollution control regulations in Beijing*, China, 2020, https://www.sciencedirect.com/science/article/pii/S1462901120313538, Last Visited: 21/9/2021

[35] –Seo Jin Ki, *A Review of Neural Networks for Air Temperature Forecasting*, 2021, https://www.mdpi.com/2073-4441/13/9/1294/htm, Last Visited: 21/9/2021

[36] – Yeong Min Jang, *Forecasting PM2.5 Concentration Using a Single-Dense Layer BiLSTM Method*, 2021, https://www.mdpi.com/2079-9292/10/15/1808/htm, Last Visited: 21/9/2021

[37] – Liqing Zhao, B. Cheng, Junliang Chen, *A Hybrid Time Series Model based on Dilated Conv1D and LSTM with Applications to PM2.5 Forecasting*, 2019, https://www.semanticscholar.org/paper/A-Hybrid-Time-Series-Model-based-on-Dilated-Conv1D-Zhao-Cheng/5ab4208e8eb941bf442e24cc71d604a9b2c65c1a, Last Visited: 21/9/2021

[38] – Yang Han, Qi Zhang, Victor O.K. Li, Jacqueline C.K. Lam, *Deep-AIR: A Hybrid CNN-LSTM Framework for Air Quality Modeling in Metropolitan Cities*, 2021, https://arxiv.org/abs/2103.14587, Last Visited: 21/9/2021

[39] – OpenAQ, https://openaq.org, Last Visited: 21/9/2021

[40] – MeteoStat, https://meteostat.net, Last Visited: 21/9/2021

[41] – World Meteorological Organization, Resolution 40, https://community.wmo.int/resolution-40, Last Visited: 21/9/2021

[42] – Python Institute, *What is Python*, https://pythoninstitute.org/what-is-python, Last Visited: 21/9/2021

[43] – python.org, *The Zen of Python*, https://www.python.org/dev/peps/pep-0020, Last Visited: 21/9/2021

[44] – Jupyter, https://jupyter.org/, Last Visited: 21/9/2021

[45] – Numpy, *what is NumPy*, https://numpy.org/doc/stable/user/whatisnumpy.html, Last Visited: 21/9/2021

[46] – Pandas, *Package Overview*, https://pandas.pydata.org/pandas-docs/stable/getting_started/overview.html, Last Visited: 21/9/2021

[47] – Matplotlib, https://matplotlib.org/, Last Visited: 21/9/2021

[48] – Seaborn, *An introduction to seaborn*, https://seaborn.pydata.org/introduction.html, Last Visited: 21/9/2021

[49] – Scikit-learn, *Getting Started*, https://scikit-learn.org/stable/getting_started.html, Last Visited: 21/9/2021

[50] – Python, datetime - *Basic date and time types*, https://docs.python.org/3/library/datetime.html, Last Visited: 21/9/2021

[51] – Python, *time - Time access and conversions*, https://docs.python.org/3/library/time.html, Last Visited: 21/9/2021

[52] – Py-openaq, http://dhhagan.github.io/py-openaq/index.html, Last Visited: 21/9/2021

[53] – Meteostat Developers, https://dev.meteostat.net/, Last Visited: 21/9/2021

[54] – TensorFlow, https://www.tensorflow.org/, Last Visited: 21/9/2021

[55] – Wikipedia, *Pearson correlation coefficient*, https://en.wikipedia.org/wiki/Pearson_correlation_coefficient, Last Visited: 21/9/2021

[56] – Wikipedia, *Spearman's rank correlation coefficient*, https://en.wikipedia.org/wiki/Spearman%27s_rank_correlation_coefficient, Last Visited: 21/9/2021

[57] – Wikipedia, *Kendall rank correlation coefficient*, https://en.wikipedia.org/wiki/Kendall_rank_correlation_coefficient, Last Visited: 21/9/2021

[58] – Franziska Bell and Slawek Smyl, 2018, *Forecasting at Uber: An Introduction*, https://eng.uber.com/forecasting-introduction/, Last Visited: 210/2021

[59] – Keras, *Bidirectional layer*, https://keras.io/api/layers/recurrent_layers/bidirectional/, Last Visited: 21/9/2021

[60] – Keras, *Dropout layer*, https://keras.io/api/layers/regularization_layers/dropout/, Last Visited: 21/9/2021

[61] – Keras, *Conv1D layer*, https://keras.io/api/layers/convolution_layers/convolution1d/, Last Visited: 21/9/2021

[62] – Paperswithcode, *Leaky-ReLU*, https://paperswithcode.com/method/leaky-relu, Last Visited: 21/9/2021

[63] – Keras, *MaxPooling1D* layer, https://keras.io/api/layers/pooling_layers/max_pooling1d/, Last Visited: 21/9/2021

[64] – Keras, *Flatten layer*, https://keras.io/api/layers/reshaping_layers/flatten/, Last Visited: 21/9/2021

[65] – Keras, *ReduceLROnPlateau*, https://keras.io/api/callbacks/reduce_lr_on_plateau/, Last Visited: 21/9/2021

[66] – Keras, *EarlyStopping*, https://keras.io/api/callbacks/early_stopping/ , Last Visited: 21/9/2021

[67] – Keras, *Adam,* https://keras.io/api/optimizers/adam/, Last Visited: 21/9/2021

[68] – Keras, *RMSprop,* https://keras.io/api/optimizers/rmsprop/, Last Visited: 21/9/2021

[69] – Keras, *Adagrad,* https://keras.io/api/optimizers/adagrad/, Last Visited: 21/9/2021

[70] – Keras, *Adadelta,* https://keras.io/api/optimizers/adadelta/, Last Visited: 21/9/2021

[71] – Keras, *Adamax*, https://keras.io/api/optimizers/adamax/, Last Visited: 21/9/2021

[72] – Keras, *Nadam*, https://keras.io/api/optimizers/Nadam/, Last Visited: 21/9/2021

[73] – Keras, *Ftrl*, https://keras.io/api/optimizers/ftrl/, Last Visited: 21/9/2021

[74] – Keras, https://keras.io/api/losses/regression_losses/, Last Visited: 21/9/2021

[75] – Keras, https://keras.io/api/metrics/regression_metrics/, Last Visited: 21/9/2021