UNIVERSITY OF PIRAEUS

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

ΔΗΜΙΟΥΡΓΙΑ ΓΝΩΣΗΣ ΒΑΣΕΙ ΔΕΔΟΜΕΝΩΝ ΜΕΓΑΛΗΣ ΚΛΙΜΑΚΑΣ ΓΙΑ ΤΗΝ ΓΝΩΣΙΑΚΗ ΔΙΑΧΕΙΡΙΣΗ ΤΗΛΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΥΠΟΔΟΜΩΝ

Αιμιλία Α. Μπαντούνα

Διπλωματούχος Τμήματος Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών

Εθνικό Μετσόβιο Πολυτεχνείο

Κάτοχος Μεταπτυχιακού τίτλου σπουδών Τμήματος Ψηφιακών Συστημάτων

Πανεπιστημίου Πειραιώς

Πειραιάς, 2015

UNIVERSITY OF PIRAEUS **ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ**

PHD DISSERTATION

KNOWLEDGE GENERATION FROM TELECOMMUNICATIONS BIG DATA FOR ENABLING COGNITIVE INFRASTRUCTURE MANAGEMENT

Aimilia A. Bantouna

B.Sc. department of Applied Mathematics and Physics

National Technical University of Athens

M.Sc. department of Digital Systems

University of Piraeus

Piraeus, 2015

UNIVERSITY OF PIRAEUS **ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ**

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Αιμιλία Α. Μπαντούνα

Πτυχιούχος Τμήματος Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών
Εθνικό Μετσόβιο Πολυτεχνείο
Κάτοχος Μεταπτυχιακού τίτλου σπουδών Τμήματος Ψηφιακών Συστημάτων
Πανεπιστημίου Πειραιώς

ΔΗΜΙΟΥΡΓΙΑ ΓΝΩΣΗΣ ΒΑΣΕΙ ΔΕΔΟΜΕΝΩΝ ΜΕΓΑΛΗΣ ΚΛΙΜΑΚΑΣ ΓΙΑ ΤΗΝ ΓΝΩΣΙΑΚΗ ΔΙΑΧΕΙΡΙΣΗ ΤΗΛΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΥΠΟΔΟΜΩΝ

Συμβουλευτική Επιτροπή:
Π. Δεμέστιχας, Καθηγητής Παν. Πειραιώς
Α. Κανάτας, Καθηγητής Παν. Πειραιώς
Α. Ρούσκας, Αναπ. Καθηγητής Παν. Πειραιώς

Εγκρίθηκε από την επταμελή Εξεταστική Επιτροπή την 23η Σεπτεμβρίου 2015,

Π. Δεμέστιχας
Καθηγητής
Πανεπιστημίου Πειραιώς

Α. Κανάτας
Καθηγητής
Πανεπιστημίου Πειραιώς

Α. Ρούσκας
Αναπληρωτής Καθηγητής
Πανεπιστημίου Πειραιώς

Μ. Θεολόγου
Καθηγητής ΕΜΠ

Μ. Αναγνώστου
Καθηγητής ΕΜΠ

Σ. Παπαβασιλείου
Καθηγητής ΕΜΠ

Γ. Βούρος
Καθηγητής
Πανεπιστημίου Πειραιώς

…………………………………

Αιμιλία Α. Μπαντούνα

Διπλωματούχος Τμήματος Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών

Εθνικό Μετσόβιο Πολυτεχνείο

Κάτοχος Μεταπτυχιακού τίτλου σπουδών Τμήματος Ψηφιακών Συστημάτων

Πανεπιστημίου Πειραιώς

Aimilia A. Bantouna

B.Sc. department of Applied Mathematics and Physics

National Technical University of Athens

M.Sc. department of Digital Systems

University of Piraeus

ABSTRACT

The continuously growing use of Internet and the optimization of the services, in terms of offering more capabilities to the users, result in the increased need for spectrum/bandwidth, a rather limited resource, and processing capabilities in core and access networks. To this end, Cognitive Radio Systems (CRSs) have been proposed for enhancing the resource allocation and utilization, and thus bridge this gap while preserving, if not enhancing, the Quality of Services (QoS) and the Quality of Experience (QoE).

Moreover, the availability of large amounts of unstructured data, which come from various sources, is seen as highly promising for deriving high level information and new insights for the business world while easier access to them through the Web facilitates the research towards this direction. However, the velocity of them being changed requires exceptional technology to efficiently process large quantities of data within tolerable timeframes. Data characterized by high volume, variety and velocity are commonly known as Big Data. These data need to be efficiently managed, handled and exploited by the Network Operators (NOs) and/or Service Providers (SPs) but human resources are not sufficient.

Knowledge building mechanisms are often proposed for addressing both of the above challenges. In particular, cognitive network management can offer solutions to the challenges posed by future networks but this requires the incorporation of knowledge that is dynamically built from its own mechanisms. Dynamically built knowledge exploits context information and allows quicker and more complex data analysis so as to better comply with the volume, the velocity and the variety of the produced Big Data. In order to build knowledge that enhances the decisions of the network, the network monitors its current state and senses information with respect to the context it functions, it collects information regarding the results of its decisions – whether the state in which it evolved allows it to have better or worse performance – and is dynamically trained to select the state with the highest performance when in similar context. During the decision making process, rules and policies of the NO and/or the SP are combined with the knowledge built from the past experience of the network so as to be respected.

To this end, this dissertation studies, designs, proposes and evaluates knowledge building mechanisms that can exploit (Big) data and enhance the decision making processes of a CRS.

Keywords: CRS, knowledge building, machine learning, Big Data, unsupervised learning

ΠΕΡΙΛΗΨΗ

Η συνεχώς αυξανόμενη χρήση του Διαδικτύου και η βελτιστοποίηση των υπηρεσιών, υπό την έννοια της προσφοράς περισσότερων δυνατοτήτων στους χρήστες, έχει ώς αποτέλεσμα την αυξανόμενη ανάγκη ραδιοσυχνότητων, μίας περιορισμένης φυσικής πηγής, και των επεξεργαστικών δυνατοτήτων των δικτύων. Τα συστήματα γνωσιακής διαχείρισης έχουν την ικανότητα να βελτιώνουν την κατανομή και την χρησιμοποίηση των πόρων ενώ παράλληλα διατηρούν, αν όχι βελτιώνουν, την ποιότητα των υπηρεσιών (QoS) και την ποιότικα της εμπειρίας των χρηστών.

Παράλληλα, η μεγάλη διαθεσιμότητα της αδόμητης πληροφορίας από διαφορετικές πηγές παρέχει την δυνατότητα της δημιουργίας γνωσης αλλά η μεγάλη ταχύτητα με την οποία η πληροφορία αυτή αλλάζει απαιτεί τέτοια τεχνολογία που να μπορεί να επεξεργάζεται μεγάλο όγκο δεδομένων σε μικρά χρονικά διαστήματα. Τα δεδομένα που χαρακτηρίζοντια από μεγάλο όγκο, ποικιλομορφία και ταχύτητα είναι γνωστά ως Big Data.

Οι μηχανισμοί δημιουργίας γνώσης αναφέρονται συχνά ώς η διέξοδος και στις 2 παραπάνω προκλήσεις των μελλοντικών δικτύων. Συγκεκριμένα, οι μηχανισμοί δημιουργίας γνώσης παράγουν δυναμικά την γνώση που περιλαμβάνει την πρότερη εμπειρία του δικτύου και μπορεί να καθοδηγήσει τις αποφάσεις του δικτύου. Συγκεκριμένα, παρακολουθούν την κατάσταση του δικτύου, συλλέγουν πληροφορίες από το περιβαλλον τους και σχετικά με την απόδοση των αποφάσεών τους και εκπαιδεύονται δυναμικά ώστε να επιλέγουν την καταλληλότερη των αποφάσεων δεδομένης της κατάστασης του δικτύου.Κατά την διαδικασία λήψης αποφάσεων για το δίκτυο, οι κανόνες και η πολιτική διαχείρισης του δικτύου συνυπολογίζονται.

Προς αυτήν την κατεύθυνση, η εν λόγω διατριβή μελετά, σχεδιάζει, προτείνει και αξιολογεί μηχανισμούς δημιουργίας γνώσης που μπορούν να αξιοποιήσουν δεδομένα μεγάλης κλίμακας και να βελτιώσει τις διαδικασίες λήψης αποφάσεων των συστημάτων γνωσιακής διαχείρισης.

Λέξεις – κλειδιά: συστήματα γνωσιακής διαχείρισης, δημιουργία γνωσης, μηχανική μάθηση, δεδομένα μεγάλης κλίμακας, μη-καθοδηγούμενες τεχνικές μάθησης

FOREWORD

The completion of this PhD dissertation was a long and difficult process, which often required both effort and total dedication. Despite the adversities and difficulties I managed to gain valuable knowledge in the field of telecommunication networks and services, since I was given the opportunity to participate in many important research projects and study significant articles in this field. None of the above would have been achieved without the actual support of many people whose contribution to my research, in various ways, was important and deserve special mention.

First of all, I would like to gratefully acknowledge my supervisor, Prof. Panagiotis Demestichas, whose dedication to this dissertation and my research work, as well as his academic experience have been invaluable to me during the accomplishment of this effort. In every sense, none of this work would have been possible without his presence. I would also like to thank my co-supervisors, Prof. Athanasios Kanatas and Associate Prof. Angelos Rouskas, who supported me and advised me at every stage of this dissertation. Moreover, I would like to thank the members of the examination committee Prof. George Vouros, Prof. Michael Theologou, Prof. Miltiades Anagnostou and Prof. Symeon Papavassiliou.

I am also grateful for the support of Dr. Kostas Tsagkaris, Assistant Prof. Vera Stavroulaki and Dr. Panagiotis Vlacheas, as well as Dr. Yiouli Kritikou, Dr. Marios Logothetis, Dr. Vassileios Foteinos, Dr. Andreas Georgakopoulos, Mr. Dimitrios Kelaidonis, Mr. George Poulios, Mr. Petros Morakos, Mr. Antonis Moustakos, Mr. Aristotelis Margaris, Mr. Ioannis Stenos and Mr. Konstintinos Petsas, for the interesting discussions we had all this time.

Of course it is needless to mention that all this effort and academic course would have been impossible to be concluded without the effective support of my family. I am forever grateful for their undivided support and understanding all these years.

Sincerely,

Aimilia A. Bantouna

ΠΡΟΛΟΓΟΣ

Η ολοκλήρωση αυτής της διδακτορικής διατριβής ήταν μια μακρά και δύσκολη διαδικασία, η οποία απαιτούσε συχνά τόσο προσπάθεια όσο και απόλυτη αφοσίωση. Παρά τις δοκιμασίες και τις δυσκολίες κατάφερα να αποκτήσω πολύτιμες γνώσεις στον τομέα των τηλεπικοινωνιακών δικτύων και υπηρεσιών, καθώς μου δόθηκε η ευκαιρία να συμμετάσχω σε πολλά σημαντικά ερευνητικά προγράμματα και να μελετήσω σπουδαία άρθρα σε αυτόν τον τομέα. Τίποτα όμως από τα παραπάνω δε θα είχε επιτευχθεί χωρίς την έμπρακτη υποστήριξη πολλών ανθρώπων των οποίων η συμβολή στην έρευνα μου, με διάφορους τρόπους, ήταν σημαντική και αξίζουν ειδικής μνείας.

Πρώτα απ 'όλα, θα ήθελα να εκφράσω την ευγνωμοσύνη μου στον επιβλέποντα μου, Καθηγ. Παναγιώτη Δεμέστιχα, του οποίου η αφοσίωση του στη παρούσα διατριβή και το ερευνητικό μου έργο, καθώς και η ακαδημαϊκή του εμπειρία ήταν πολύτιμη για μένα κατά τη διάρκεια της υλοποίησης αυτής της προσπάθειας. Τίποτα δε θα ήταν δυνατό χωρίς την παρουσία του. Θα ήθελα να ευχαριστήσω τους συνεπιβλέποντες, Καθηγ. Αθανάσιο Κανάτα και Αν. Καθηγ. Άγγελο Ρούσκα, οι οποίοι με στήριξαν και με συμβούλεψαν σε κάθε στάδιο αυτής της διατριβής. Επιπλέον, θα ήθελα να ευχαριστήσω τα μέλη της εξεταστικής επιτροπής Καθηγ. Γεώργιο Βούρο, Καθηγ. Μιχαήλ Θεολόγου, Καθηγ. Μιλτιάδη Αναγνώστου και Καθηγ. Συμεών Παπαβασιλείου.

Είμαι επίσης ευγνώμων για την υποστήριξή τους και θα ήθελα να ευχαριστήσω τους Δρ. Κώστα Τσαγκάρη, Επίκ. Καθηγ. Βέρα Σταυρουλάκη και Δρ. Παναγιώτη Βλαχέα, καθώς και τους Δρ. Γιούλη Κρητικού, Δρ. Μάριο Λογοθέτη, Δρ. Βασίλειο Φωτεινό, Δρ. Ανδρέα Γεωργακόπουλο, Δημήτρη Κελαιδώνη, Γιώργο Πούλιο, Πέτρο Μοράκο, Αντώνη Μουστάκο, Αριστοτέλη Μάργαρη, Ιωάννη Στενό και Κωνσταντίνο Πέτσα για τις ενδιαφέρουσες συζητήσεις που είχαμε όλο αυτό το διάστημα.

Φυσικά, είναι περιττό να αναφέρω ότι όλη αυτή η προσπάθεια και ακαδημαϊκή πορεία θα ήταν αδύνατο να ολοκληρωθεί χωρίς τη στήριξη της οικογένειάς μου. Είμαι για πάντα ευγνώμων για την αμέριστη υποστήριξή τους και την κατανόηση όλα αυτά τα χρόνια.

Με εκτίμηση,

Αιμιλία Α. Μπαντούνα

Table of Contents

List of Figures

List of Tables

1 INTRODUCTION

1.1 Research Area - Motivation

Rapid evolution of wireless communications has led to new wireless technologies, each one demanding its own piece of electromagnetic radio spectrum. However, this source is naturally limited. Moreover, despite the large recent research initiatives that target a more flexible and automated resource management, current static spectrum assignment often leads to its underutilization while current resource management is rather manual and quite static. In particular, resources are planned based on the worst case scenario, i.e., the most demanding scenario. Furthermore, the users' mobility increases. Overall, the continuously changing environment often results in calling for reconfigurations at various time scales. Thus, the dynamicity that the network should handle increases as well.

Accordingly, the deployment of technologies which have the ability to explore and use the underutilized bands of frequency and the challenges of future networks is a sine qua non. Research performed in this field during the last decade has proposed both centralized and distributed dynamic spectrum assignment techniques. CRSs [1][2] can further enhance such techniques by adding past experience and knowledge to be taken into account during spectrum assignment.

CRSs are capable of intelligently adapting to the highly varying and disparate modern environments [1][2]. In particular, they have the ability to adjust their operation according to the external, environmental stimuli, the demands of the users/applications and their past experience. Based on this ability, future cognitive radio systems will be able to change their parameters (carrier frequency, radio access technology, transmit power, modulation type etc), observe the results and decide which is the best combination of those parameters in order to get into a better operational state. The above is an iterative process, well known as "cognition cycle". A typical cognitive cycle consists of three cooperative phases [2][3]. During, the first phase, known as "radio scene analysis", the system collects measurements from its environment (e.g. conditions related to interference) and explores different configurations. In the second phase, "channel estimation and predictive modelling", the output of the first phase is used for discovering the capabilities of each candidate configuration, wherein past experience of the system may also be used. Finally, in the last phase, known as "configuration selection", the system decides for the best configuration and accordingly adjusts its operation parameters.

However, the cognition cycle is often proved to comprise very arduous and time-consuming processes especially due to the large number of configurations which need to be investigated. In these terms, machine learning techniques are expected to be beneficial for speeding up the whole process.

Moreover, according to [4], unstructured data increase more than 50 percent every year due to both numerous available applications and user devices. In particular, the pervasiveness of the network technologies in everyday life through e.g., social networking, public information, Rich Site Summary (RSS, also known as Real Simple Syndication) feeds, and other applications for informative purposes, payments, communication, social media, Internet of Things (IoT), etc., and the advanced capabilities of terminals/users' devices create a vast amount of diverse data and result in the exponential increase of available digital data. Furthermore, the large number of disparateries, in software and hardware, user devices (such as laptops, notebooks, mobiles and others) further increase the volume, the variety, and the velocity of the available digital data making their management difficult with on-hand database management tools.

These data have been referred to as "Big Data", and despite their difficulty being managed, they have been conceptualized to offer valuable insight on application provisions and network technology aspects when they get analyzed and properly processed. The heavy data analysis that is required for retrieving these insights suggests that new architectures and mechanisms are needed for (a) handling the volume of data that will be stored, (b) aggregating, (c) exploiting, and (d) building knowledge on them. In telecommunication networks, the term "knowledge", refers to information useful for the operators and the network, which cannot be directly monitored. Following the "knowledge" definition of the European Committee for Standardizations [5] and specifically, the "Official Guide to Good Practice in Knowledge Management", "knowledge is the combination of data/information with the opinions/skills/experience of experts, which can be humans or computational systems and results in a valuable asset that can be used to aid decision making".

To this end, autonomic processes and techniques that allow the management, the analysis and the exploitation of Big – in terms of volume, velocity and variety – data for supporting cognitive network management decisions through e.g., alarms regarding the future states of the network are required.

CRSs are able of exploiting the past experience of the network and make decisions that optimize its functions and capabilities. Optimization and machine learning techniques are used towards this direction. In particular, machine learning techniques are used for developing mechanisms that process raw data collected through sensing so as to produce elaborated data and alarms and build knowledge with respect to the past experience of the network (knowledge building mechanisms). NOs and/or autonomic decision making mechanisms can then exploit this information in order to better (in terms of speed and efficiency) manage network through either reactive or proactive diagnosis of upcoming – potentially undesired – network states.

1.2 Dissertation's Contribution

This dissertation deals with the "Knowledge Generation from telecommunication Big Data for enabling Cognitive Infrastructure Management". Therefore, it involves the study of

- the framework where knowledge building mechanisms can be hosted;
- the data, both sensed by the network and human-oriented, that can add value to the generation of the knowledge and for enhanced and past-experienced-based decisions;
- machine learning techniques that can support the development of knowledge building mechanisms;
- the design of the knowledge building mechanisms per se (using mainly unsupervised machine learning techniques); and

the experimental validation of the knowledge building mechanisms with the respective performance results.

1.3 Dissertation Structure

The dissertation is structured in chapters, each of which provides a detailed description on the research activities performed with regards to the topics noted in Section 1.2. A brief description of them follows in the next paragraphs.

Chapter 2 This thesis, deals with the development of knowledge building mechanisms so as to enhance the cognitive processes of future networks. The objective is to provide realistic and deployable solutions, thus both the framework that hosts the mechanisms and the data upon which the mechanisms will be built need to be defined. Chapter 2 describes the placeholder of these mechanisms within the Generic Autonomic Network Architecture (GANA), as described from the European Telecommunications Standards Institute (ETSI) for Autonomic network engineering for the self-managing Future Internet (AFI), and the data that can be sensed by the network and be recorded in current and future Management Information Bases (MIBs) of Evolved Universal Terrestrial Radio Access Network (E-UTRAN) nodes, as defined in Third Generation Partnership Project (3GPP) specification documents.

Research on the proposed solution, as it is described in this chapter, resulted in the following publication:

- A. Bantouna, K. Tsagkaris, P. Demestichas, "Knowledge Functional Block for E-UTRAN", Accepted to 5th IEEE International Workshop on Management of Emerging Networks and Services (IEEE MENS 2013), Atlanta , GA USA, Dec. 2013

Chapter 3 This chapter presents the categorization of the offered machine learning techniques, according to the training method that is used in order to extract knowledge out of raw data, and provides the mathematical background for four of them (one example per category).

Research on the proposed solution, as it is described in this chapter, resulted in the following publication:

- A. Bantouna, K. Tsagkaris, V. Stavroulaki, P. Demestichas, G. Poulios, "Machine Learning applied to Cognitive Communications", Cognitive Communications: Distributed Artificial Intelligence (DAI), Regulatory Policy & Economics, Implementation. H. Zhang and D. Grace, J. Wiley and Sons, Ltd, Chichester, UK, Print ISBN: 9781119951506 (Oct. 2012), Online ISBN: 9781118360316 (Jul. 2012), chapter 6, doi: 10.1002/9781118360316.ch6
- A. Bantouna (presenter), K. Tsagkaris, V. Stavroulaki, P. Demestichas, "Machine Learning Techniques for Autonomic/ Cognitive Networking", in the context of the 7th International Conference on Network and Service Management CNSM 2011, held on 24-28 October 2011, in Paris, France

Chapter 4 This chapter aims at providing comparative studies of similar knowledge building mechanisms which are based on supervised and unsupervised machine learning techniques. In these studies, the advantages and the disadvantages of the developed mechanisms are analyzed while emphasis is also put on their complementarity. The developed mechanisms target at building knowledge and providing insights with respect to the capabilities of the network.

The work described in this chapter is presented in the following publications:

- K. Tsagkaris, A. Bantouna, P. Demestichas, "Self-Organizing Maps for Advanced Learning in Cognitive Radio Systems", Computers & Electrical Engineering, Elsevier, Vol. 38, No. 4, p. 862–881, July 2012, http://dx.doi.org/10.1016/j.compeleceng.2012.03.008
- A. Bantouna, V. Stavroulaki, Y. Kritikou, K. Tsagkaris, P. Demestichas, K. Moessner, "An overview of learning mechanisms for cognitive systems", Published to EURASIP Special Issue on Ten Years of Cognitive Radio: State of the Art and Perspectives, EURASIP Journal on Wireless Communications and Networking 2012, 2012:22 doi:10.1186/1687-1499-2012-22, January 2012

Chapter 5 Chapter 5 mechanisms address the challenge of analysing data related to the traffic of either a core or an access wireless network. In particular, the first mechanism is building knowledge on the traffic of a core network and predicts how possible it is to run on a congested link. The data exploited in this mechanism are directly monitored by the network, i.e., refer to network parameters, and thus allow narrow timeslots for proactively overcoming a congested link. On the other hand, the second mechanism builds knowledge with respect to the traffic of the access network but takes into account human-oriented parameters (e.g., time, date, location, etc.) as well, allowing to also foresee more long-term situations and adjust the network parameters accordingly.

Research in this field resulted in the following publications:

- A. Bantouna, G. Poulios, K. Tsagkaris, P. Demestichas, "Network Load Predictions based on Big Data and the Utilization of Self-Organizing Maps", JNSM Special Issue 2013 : Springer Journal of Network and Systems Management – Special Issue on "Data Mining for Monitoring and Managing Systems and Networks", Sept. 2013, DOI: 10.1007/s10922-013-9285-1, Volume 22, Issue 2 (2014), Page 150-173
- A. Bantouna, K. Tsagkaris, V. Stavroulaki, G. Poulios, P. Demestichas, "Learning Techniques for Context Diagnosis and Prediction in Cognitive Communications", Cognitive Communications: Distributed Artificial Intelligence (DAI), Regulatory Policy & Economics, Implementation. H. Zhang and D. Grace, J. Wiley and Sons, Ltd, Chichester, UK, Print ISBN: 9781119951506 (Oct. 2012), Online ISBN: 9781118360316 (Jul. 2012), chapter 9, doi: 10 10.1002/9781118360316.ch9
- M. Ghader, A. Bantouna, L. Bennacer, G. Calochira, B. Fuentes, G. Katsikas, Z. Yousaf, "On Accomplishing Context Awareness for autonomic network management", accepted at Future Network and Mobile Summit 2012, 4 - 6 July 2012, Berlin, Germany
- A. Bantouna, G. Poulios, K. Tsagkaris, P. Demestichas, "Dynamic Management of Cognitive Radio Networks using Big Data", 3rd ACROPOLIS Workshop and Industry Panel, London, Sept. 2013

Chapter 6 This chapter examines if and how a knowledge building mechanism can be applied on the transport layer and enhance existing mechanisms. In particular, the study presented here, proposes a knowledge building mechanism that could support the functionality of the TCP Vegas congestion avoidance mechanism and enhance its decisions with respect to the selected congestion window. The selection of a congestion window closer to the needs of the network can in turn minimize the instabilities caused to the network utilization when trying to avoid loss of information due to congestion.

The outcome of this research, as it is described in this chapter, resulted in the following publication:

- A. Bantouna, K. Tsagkaris, G. Poulios, A. Manzalini, P. Demestichas, "Knowledge in Support of Congestion Control Mechanisms", poster, Future Network Mobile Summit (FuNeMS) 2012, 4 - 6 July 2012, Berlin, Germany

Chapter 7 Last but not least, during this thesis, discussions related to the acceptability of the mechanisms and how humans can trust the network to make the right decisions also took place. To this end, chapter 7 proposes a mechanism based on reinforcement learning that is envisioned to measure the performance of autonomic loops and evaluate how trustworthy their proposed actions are.

The outcome of this research, as it is described in this chapter, resulted in the following publication:

- L. Ciavaglia, S. Ghamri - Doudane, M. Smirnov, P. Demestichas, V. Stavroulaki, A. Bantouna, Unifying Management of Future Networks with Trust, Bell Labs Technical Journal (BLTJ), Special Issue On "Delivering Network Assurance through Secure and Reliable Products, Software, Services and Solutions", Vol. 17, No. 3, p. 193-212, December 2012

Chapter 8 The last chapter discusses the main aspects introduced by this dissertation. Furthermore, on-going challenges are noted and finally the dissertation is concluded.

1.4 Chapter References

[1] J. Mitola III, Jr. G. Maguire, “Cognitive radio: making software radios more personal”, IEEE Personal Commun., Vol. 6, No 4 , pp13 -18, 1999

[2] S. Haykin, “Cognitive radio: brain-empowered wireless communications, IEEE Journal on Selected Areas In Communications”, Vol. 23, No. 2, pp. 201-220, 2005

[3] P. Demestichas, A. Katidiotis, K. Tsagkaris, E. Adamopoulou, K. Demestichas, “Enhancing Channel Estimation in Cognitive Radio Systems by means of Bayesian Networks”, Wireless Personal Communications, vol. 49, Issue 1, pp 87-105, 2009

[4] Lohr, S.: The Age of Big Data. New York Times. http://www.nytimes.com/2012/02/12/sunday-review/big-datas-impact-in-the-world.html?_r=1 (2012). Accessed 20 February 2012

[5] European Committee for Standardization Website. http://www.cen.eu/cen/News/Newsletter/Pages/default.aspx. Accessed 17 October 2012

2 STRUCTURES AND MODELS OF KNOWLEDGE VISUALIZATION

2.1 Chapter Outline

Large availability of digital data, or else Big Data, offers to data scientists the chance of exploiting them and building knowledge on different aspects of everyday life related to e.g., people preferences and habits, transportation, health systems, telecommunications etc. This study targets to exploit this opportunity in the telecommunications era. Towards this direction, the chapter presents the architectural aspects, the role and the foundation of the functionalities of the knowledge functional block. Knowledge functional block comprises of knowledge building mechanisms that learn different aspects of future wireless networks in order to enhance their decisions and thus their functionality. These aspects may involve context, efficiency of the decisions, energy efficiency, trust and others. Data that can be directly monitored from an E-UTRAN are identified and modeled to guide the input that can be used for the knowledge building mechanisms of the block. The keywords for this chapter are: Big Data, Knowledge, E-UTRAN, MIBs.

2.2 Introduction

Technology intervention in humans' everyday life has much facilitated our daily routine and communications. On the other hand, the generated data from Internet transactions, emails, videos, click streams, and/or all other digital sources available today and in the future are rapidly increasing in size and disparity. Technology enhancements that have made feasible all these application to be hosted in a large number and type of user devices, such as smart phones, laptops and other, and the increase of instruments and sensors for monitoring the automatic processes of the everyday life (i.e., sensors in e.g., our industrial equipment, automobiles, electrical meters and shipping crates) have also contributed to this.

Data scientists used to analyze themselves the available data, identify the trend of the market, build Business Intelligence (BI) and propose the next actions of the business in which they are employed. However, current explosion of data availability makes their analysis from humans difficult. This large availability of digital data, in terms of volume, variety and velocity (in which they change), is commonly stated as Big Data. Their volume currently reaches the order of petabytes, exabytes, and zettabytes, they may refer to medical imaging, gene sequencing, video surveillance, social media, smart electrical grids, mobile phone sensors and others [1] and forces data analysts to exploit tools such as machine learning, data mining, data visualization, etc., in order to build knowledge on them [2]. Therefore, although Big Data raise concerns with respect to their storage, handling and managing, they also promise to offer insights, BI and business opportunities [2][3].

On the other hand, networks complexity and dynamicity have increased to such an extent that the current policy-based reconfiguration with minor changes is not enough. Two ways out have been identified: a) to turn to manual tuning performed by the operators and b) to exploit autonomic network management. The first case turns out to be labor-intensive, expensive and intrinsically error-prone due to both the complexity of the networks and their dynamicity which requires often reconfigurations [4].

The case of autonomicity, however, promises to dynamically tune the network based on operators' policies, i.e., targets and rules, providing the means to self-analyze the changes in the network without requiring human intensive efforts. The latter can further be translated to reduced Operational (OPEX) and Capital Expenditures (CAPEX) for the operators. For example, autonomically enforcing an energy-efficient solution where/ when circumstances allow it (e.g., reduced number of users to be served) certainly impacts OPEX. Accordingly, an autonomic network that is dynamically and online self-configured based on the users' demand requires less often capital investments (e.g., to increase the number of its antennas) comparing to networks which are not dynamically reconfigured and their initial configuration usually reflects the worst case scenarios.

Autonomic network management consists of the following functions usually referred to with the acronym MAPE [5]:

- Monitor: for probing, aggregating, filtering and reporting the operating parameters of the managed entities;

- Analyze: for processing, building knowledge and modeling the operating environment;
- Plan: for reasoning and deciding the next actions according to network policies; and
- Execute: for taking actions and reconfiguring the network based on the plan designed in the previous function.

This chapter focuses on the second function, i.e., the "Analyze" function which is very important for designing the next steps and actions of the network, especially in a highly dynamic environment as the one expected in future networks. More specifically, "Analyze" function receives data from the "Monitor" function and exploits them to build knowledge and model the operating environment. This knowledge is then provided as input to the "Plan" function so as the latter to reason and select among the available solutions the most appropriate. Eventually, the actions that comprise the plan are executed by the last function.

Each of these functions plays an important role for the autonomic loops of a network management framework but the real autonomicity is interwoven with knowledge, i.e., the capability of the network to learn the patterns and the models of its environment which will then guide/ consult "Plan" function. In particular, knowledge is envisaged to enhance decision making in terms of speed, efficiency and context-awareness and thus, facilitate dynamic network reconfigurations. Big Data offer the opportunity and the necessary, multi-oriented information for building this required knowledge.

The following sub-sections familiarize the reader with the exact problem that will be studied, describe where the knowledge functional block stands from an architectural point of view, analyses its envisaged functionalities and presents the data that can be monitored in an E-UTRAN and be used by the knowledge building mechanisms of the block.

2.3 Problem statement

The study focuses on "Analyze" function of MAPE and specifically on knowledge building processes. Towards this direction, it introduces the knowledge functional block and the mechanisms that comprise it. The mechanisms:

a) exploit the data that can be monitored in an Evolved Universal Terrestrial Radio Access Network (E-UTRAN),
b) learn the way they are connected to each other,
c) identify their pattern and build knowledge on various networking aspects.

The built knowledge will then be available to the "Plan" function, i.e., the decision making mechanisms of the autonomic management framework, and will be provided to them when requested.

2.4 Architectural Aspects

From the architectural point of view, the functional block is compliant with GANA [6], i.e., the architectural reference model for autonomic networking, cognitive networking and self-management proposed by AFI ETSI. A simplified overview of some of the key aspects of GANA is depicted in Figure 2.1. GANA is based on four levels of Decision-making-Elements (DEs) that instrument the network elements which collaboratively work together. In particular, the four levels of DEs (in descending order) are: a) the Network Level DEs, b) the Node Level DEs, c) the Function Level DEs and d) the Protocol Level DEs. Each DE manages one or more, lower level DEs through autonomic control loops that send them commands, objectives and policies, while they receive feedback from them in the form of monitored data or knowledge. Moreover, Node Level DEs (one per node) are responsible for orchestrating the Function Level DEs of the network elements while Network Level DEs perform the same task for the Node Level DEs. Network Level DEs, complemented by the Overlay Network for Information eXchange (ONIX), i.e., a distributed scalable system of information servers, and the Model-Based-Translation Service (MBTS), an intermediation layer between the Knowledge Plane and the Network Elements for the purpose of translating information and commands/responses, are also the Functional Blocks, i.e., the groups of protocols and mechanisms, of the Knowledge Plane.

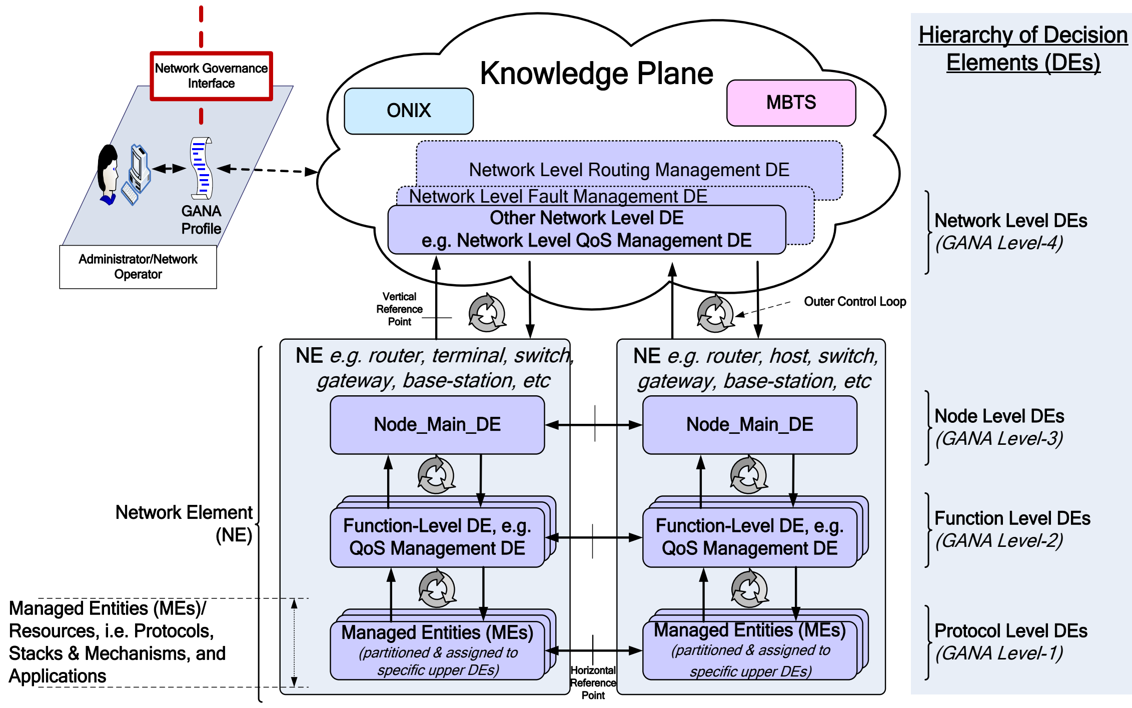


Figure 2.1: Simplified overview of the Functional Blocks of GANA [6].

The management of the network by the operator and/or the network administrator is made through GANA profiles which comprise operators' policies, high-level objectives and some configuration-data. GANA profiles are eventually translated in the knowledge plane into instructions for the Node, Function and Protocol Level DEs.

The proposed functional block is envisaged to be incorporated in the Knowledge Plane of the GANA architecture. It consists of different mechanisms which build knowledge on the past experience of the network and can thus facilitate the translation of the GANA profiles from the Network Level DEs into the commands issued to the lower level DEs for enforcement, complementing this way the operators' policies and high-level network objectives. Network operator will be able to monitor the provided (by the knowledge building mechanisms) knowledge at any time through the Graphical User Interface (GUI) of the functional block.

2.5 Functional Role

The proposed knowledge block is a logical group of multiple/ different mechanisms that will build knowledge on different aspects of the network such as:

- the possibility of a context c to be encountered, e.g., what will the load of the network be?,
- user preferences/ habits, e.g., what is the mobility of the users? What is their preferred QoS for application a?, etc.
- the efficiency of a decision d, e.g., how efficient was decision d in terms of energy consumption or spectrum allocation?
- alternative decisions that can be used when context c is encountered
- contradicting policies, etc.

Each mechanism will reflect a different aspect/ problem and, through the knowledge block interfaces, will have access to databases which will include both network monitored data and more human-oriented ones, such as the user preferences or the date or even environmental data. Therefore, the output of the knowledge block is the knowledge/ response to these aspects while the inputs are all these data needed, depending on the targeted problem, for building the knowledge (see also section 2.6).

Figure 2.2 depicts the GUI of the proposed functional block. Each button allows the user to select the knowledge mechanism he is interested in and reveals the GUI of the selected mechanism. In this GUI, the user is able to select the input parameters according to which the knowledge on the specific aspect will be built, the learning technique according to which this knowledge will be produced (when more than one learning techniques are offered) and algorithmic-specific parameters.

Knowledge will be built using machine learning techniques depending on the knowledge and the data to be used. Those may be:

a) supervised, such as Neural Networks (NNs) [7][8] and Bayesian statistics [8][9], i.e., techniques which use the desired outcome in order to guide the algorithm (during its training) what the output should be with respect to the input;
b) reinforcement learning like Q-learning [10][11][12], i.e., techniques that “award” the system during their training when it comes up with the correct answer; or

c) unsupervised, e.g. Self-Organizing Maps [13][14], i.e., techniques which use neither the desired output nor an award during their training. On the contrary, they are able to identify the patterns of the data and exploit them in order to conclude the output.

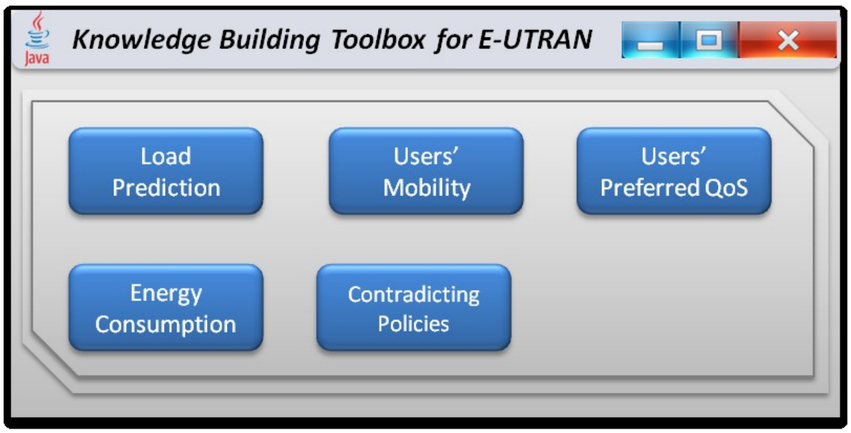


Figure 2.2: Graphical User Interface (GUI) of the Knowledge Functional Block.

2.6 Foundation for the functionality

The foundation of the functional block consists of the parameters that can be extracted by the infrastructure. Towards this direction, the study is initiated by the identification of data that are or will be offered by today and future MIBs of E-UTRAN nodes. In particular, 3GPP specification documents [15]-[21] were studied to identify the performance measurements of an E-UTRAN that are related to three specific issues: a) the load of the network, b) the mobility of the users and c) the user experience/ satisfaction.

2.6.1 Load of the network

For the network load, the identified measurements referred to eNodeBs (eNBs), Donor eNBs (DeNBs) and Relay Nodes (RNs), and more specifically to the setup, release, activity and number of their E-UTRAN Radio Access Bearers (E-RABs), to the bit-rate, the active User Equipments (UEs) and the throughput and to the Radio Resource Utilization (RRU).

The most common way of monitoring the load of the network are the measurements related to the average and/ or maximum downlink/ uplink bit-rate (PdcpSduBitrateDl, PdcpSduBitrateUl, PdcpSduBitrateDlMax, PdcpSduBitrateUlMax) and the IP throughput (IPThpDl/ IPThpUl) over time. However, other measurements are offered as well depending on the purpose of use. Adding the number of the successfully established initial (EstabInitSuccNbr), additional (EstabAddSuccNbr) and for incoming handovers (EstabInHoSuccNbr) E-RABs - per Quality Channel Indicator (QCI) or not (sum suffix) - reveals the load of the network in terms of E-RABs. Combining the average (UsageNbrMean) and maximum (UsageNbrMax) number of simultaneous E-RABs - in total or per QCI - with time allows the network administrator to perceive the pattern of the load in terms of E-RABs over time. When it comes to services that use Random Access Channels (RACH), the mean number of received low range (group A) or high range (group B) RACH preambles per second can be exploited instead (RachPreambleAMean and RachPreambleBMean). Another, more user-centric, approach for estimating the load of the network is the average number of active UEs in the downlink (UEActiveDl) and/ or the uplink (UEActiveUl). These measurements can also be given either in groups of QCIs or in total. In case the network operator/ administrator monitors the load of the network so as to adjust the available Physical Resource Blocks (PRB) of the eNodeB (eNB), then the percentage of the used downlink and/ or uplink PRBs for traffic (PrbDl/ PrbUl) or for any reason (PrbTotDl/ PrbTotUl) are the most appropriate measurements. These measurements when offered for RNs are indicated as PrbDlRN, PrbUlRN, PrbTotDl and PrbTotUl, respectively, while they are all separated depending on the respective QCI. Similar measurements are also offered when the load is monitored in terms of the time that the resources are used, i.e., the maximum time an E-RAB needs to be setup in ms (EstabTimeMax), the in-session activity time for the UE (SessionTimeUE) or the E-RABs (SessionTimeQCI), the percentage of time when all dedicated RACH preambles are used (RachDedicatedPreamblesAssigned) and the percentage of time during which all available PRBs for traffic on the downlink and/ or the uplink have been assigned to UEs (PrbCongestionDl, PrbCongestionUl). Finally, the number of E-RABs requested by eNBs/ RNs to release due to high load (RelEnbNbr) and the peak processor usage (PeakProcessorUsage), given the fact that high load requires more processing from the system, indirectly imply the load of the network as well.

The above mentioned available measurements for an E-UTRAN are depicted (in compliance to their categorization by the 3GPP specification document [19]) in Figure 2.3.

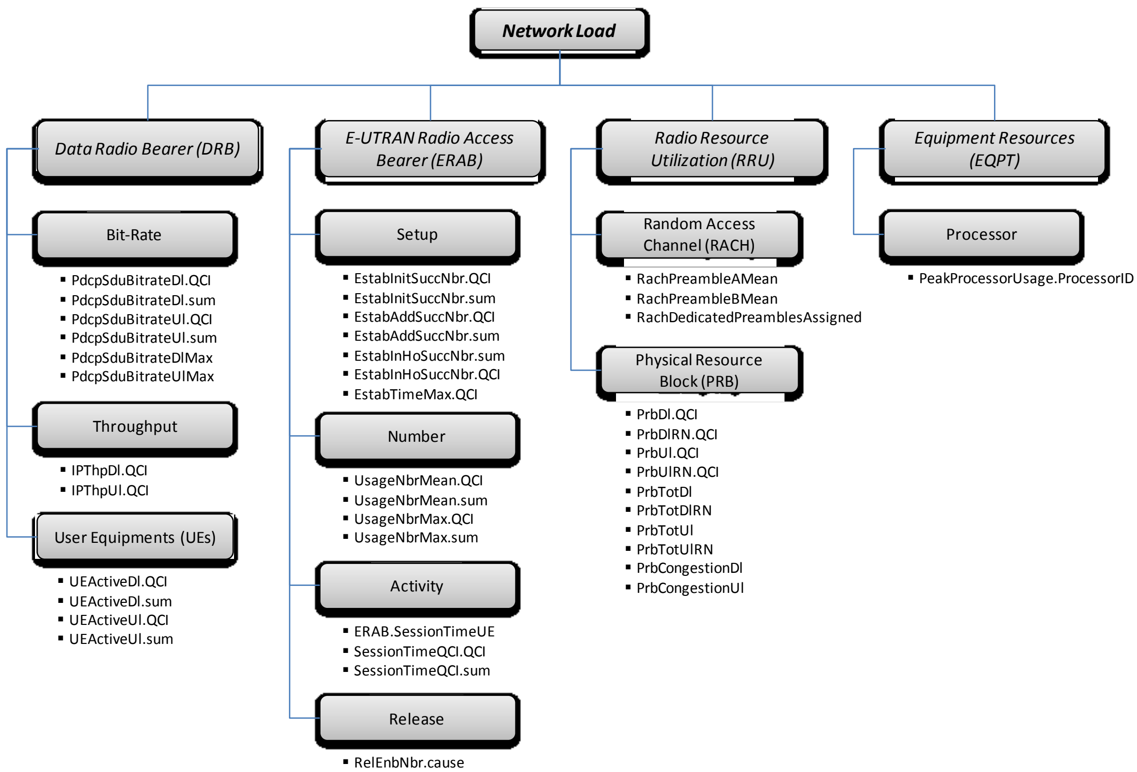


Figure 2.3: Model of data that can be retrieved by an E-UTRAN MIB and be used to build knowledge on the network load

2.6.2 Mobility of the users

The mobility of the users is represented by the handovers of the connection of their UEs from cell to cell. In particular, as the user moves in the space and his distance from the node changes, his connection needs to be transferred in another cell. Exceptions such as the case where the user's connection to the network changes due to e.g., congestion of the cell need to be taken into account. Depending on the characteristics of the network and the operating parameters of the cells, the respective measurements may refer to intra- and/ or inter-RAT handovers, among the same or different eNBs/ RNs and frequencies. The measured handovers do not necessarily need to be successful since even the attempt for a handover is enough to showcase the need and thus the change of the user's position with respect to the cell. Overall, the following measurements can be combined with - at least - time to build knowledge on and identify the pattern of the mobility of the users over time:

- Attempted outgoing intra-eNB/RN handovers per handover cause (HO.IntraEnbOutAtt.Cause);
- Attempted outgoing intra-DeNB handover preparations from DeNB cell to RN per handover cause (HO.IntraDenbOutPrepToRnAtt.Cause);
- Attempted outgoing inter-eNB handover preparations (HO.InterEnbOutPrepAtt);
- Attempted outgoing handovers per handover cause and LTE target cell specific (HO.OutAttTarget.Cause);
- Attempted outgoing intra-frequency handovers (HO.IntraFreqOutAtt);
- Attempted outgoing inter-frequency handovers - gap-assissted measurement (HO.InterFreqMeasGapOutAtt);
- Attempted outgoing inter-frequency handovers - non gap-assisted measurement (HO.InterFreqNoMeasGapOutAtt);
- Attempted preparations of outgoing handovers to the cells outside the RN (HO.OutRNOutPrepAtt); and
- Attempted outgoing inter-RAT handovers per handover cause (HO.IartOutAtt.Cause).

Figure 2.4 depicts these measurements categorized based on [19].

2.6.3 User experience/ satisfaction

User experience/ satisfaction is commonly stated as QoE, i.e., how the users perceive the quality of the offered services. The most often used parameters by the operators to deduce QoE are the delay (PdcpSduDelayDl), the drop rate (PdcpSduDropRateDl), i.e., the rate of dropped packets before leaving the node due to e.g., congestion, and the loss rate (PdcpSduAirLossRateDl and PdcpSduAirLossRateUl for the downlink and the uplink respectively), i.e., the rate of packets transmitted but not received to the destination. All three parameters are expected to be monitored in an E-UTRAN and their measurements can be provided both separated per QCIs and in total. Apart from these performance measurements, the length of time that the cell is unavailable for each cause or in total (CellUnavailableTime), the number of released active E-RABs (RelActNbr) and the CQI as reported by the UEs in the cell (WBCQIDist) can enhance operators' point of view.

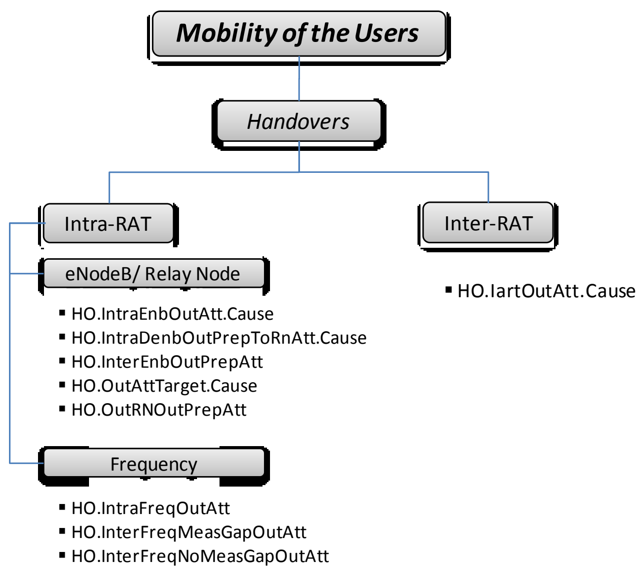


Figure 2.4: Model of data that can be retrieved by an E-UTRAN MIB and be used to build knowledge on the mobility of the users.

Finally, valuable insights with respect to the QoE can be provided when combining the performance measurements of the number of initial (EstabInitAttNbr), additional (EstabAddAttNbr) and for incoming handovers attempts (EstabIncHoAttNbr) for E-RABs establishment (per QCI or in total) with the successful ones (EstabInitSuccNbr, EstabAddSuccNbr, EstabInHoSuccNbr) in order to calculate the success rate of E-RAB setups. The pattern of the mean (EstabTimeMean) and maximum (EstabTimeMax) E-RAB setup time over time offer an additional hint in this case. For RACH specific services, the RACH access delay (RachAccessDelayDist) can be used instead.

Figure 2.5 depicts the most relevant performance measurements of an E-UTRAN for the experience of its users.

2.7 Conclusions

This chapter introduced a study according to which the large availability of Big Data can be exploited in order to build knowledge on different aspects of telecommunications (context, efficiency of decisions, user preferences, etc.). In particular, the knowledge functional block is proposed to host different mechanisms which learn the pattern of information that can facilitate the dynamic management of an E-UTRAN. The functional block is positioned to GANA architecture and its functional design has been described. Moreover, the chapter identifies and presents the measurements that can be monitored in an E-UTRAN and be exploited as inputs to knowledge building mechanisms related to the network load, the users' mobility and the users' preferences/ satisfaction. The identified measurements are compliant to the 3GPP specifications for E-UTRANs in order to ensure the credibility of the study.

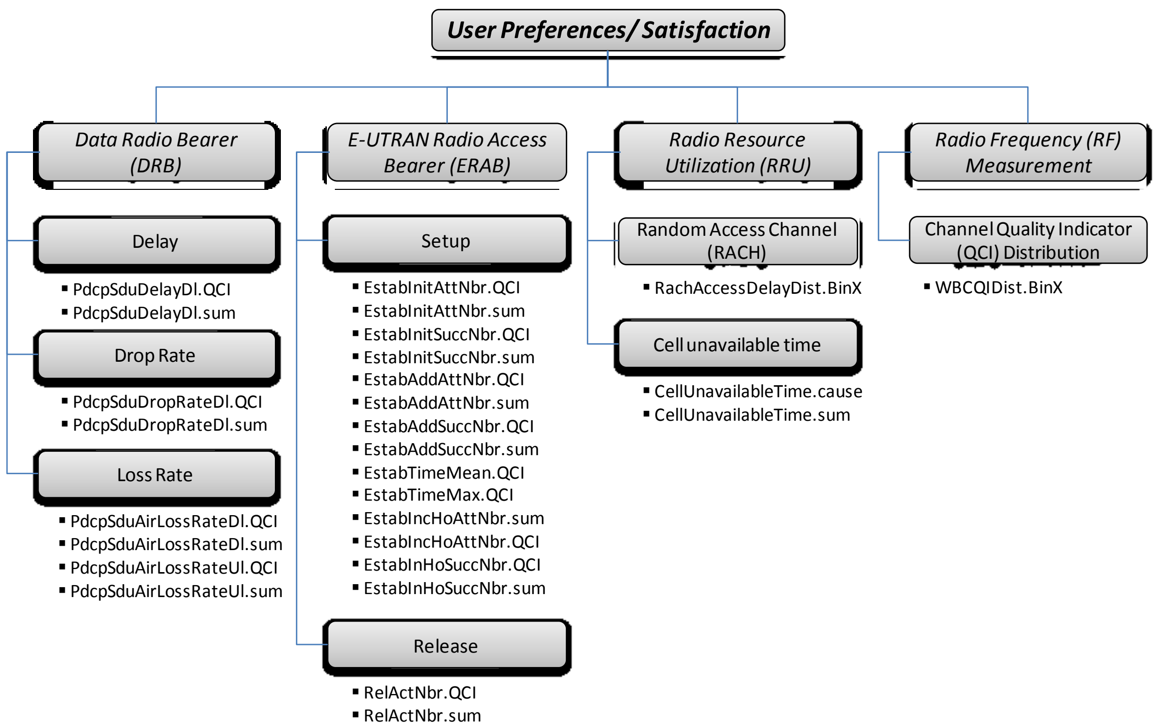


Figure 2.5: Model of data that can be retrieved by an E-UTRAN MIB and be used to build knowledge on the user experience/ satisfaction.

The next chapters of this dissertation include a) the expansion of research for measurements related to other problems as well, e.g., energy efficiency, network performance and stability, b) the identification of data that do not come from the telecommunication area but can be exploited for enhancing the built knowledge of the mechanisms, e.g., environmental and user data, c) the identification of appropriate machine learning techniques with respect to each targeted problem, d) the design, building and evaluation of the knowledge building mechanisms, and e) the validation of the "knowledge functional block for E-UTRAN" with respect to its functionality and its compliance to network operators preferences.

2.8 Chapter References

[1] EMC2, "EMC Annual Report", available at http://www.emc.com/corporate/annual-report/big-data.htm?cmp=bac_big_data_brand_forbesq22013 [last accessed: June 2013]

[2] J. Manyika, M. Chui, J. Bughin, B. Brown, R. Dobbs, C. Roxburgh, A. Hung Byers, "Big Data: The next frontier for innovation, competition, and productivity.", McKinsey Global Institute, May 2011, Online at http://www.mckinsey.com/Insights/MGI/Research/Technology_and_Innovation/Big_data_The_next_frontier_for_innovation [last accessed: June 2013]

[3] S. Lohr, "The Age of Big Data", New York Times, February 11th, 2012, available online at http://www.nytimes.com/2012/02/12/sunday-review/big-datas-impact-in-the-world.html?_r=1 [last accessed: June 2013]

[4] R. Riggio, F. D. Pellegrini, D. Miorandi, I. Chlamtac, "A Knowledge Plane for Wireless Mech Networks", Ad-Hoc & Sensor Wireless Networks, Vol. 5, pp. 293-311, 2007

[5] "Practical autonomic computing: Roadmap to self managing technology," IBM, Tech. Rep., 2006.

[6] European Telecommunications Standards Institute (ETSI) Group Specification (GS) for Autonomic network engineering for the self-managing Future Internet (AFI), "Autonomic network engineering for the self-managing Future Internet (AFI); Generic Autonomic Network Architecture (An Architectural Reference Model for Autonomic Networking, Cognitive Networking and Self-Management)", ETSI GS AFI 002 V1.1.1 (2013-04)

[7] A. K. Jain, J. Mao, K. M. Mohiuddin, "Artificial Neural Networks: A Tutorial", Computer, pp. 31-44, 1996

[8] A. Bantouna, K. Tsagkaris, V. Stavroulaki, P. Demestichas, G. Poulios, "Machine Learning applied to Cognitive Communications", Cognitive Communications: Distributed Artificial Intelligence (DAI), Regulatory Policy & Economics, Implementation. H. Zhang and D. Grace, J. Wiley and Sons, Ltd, Chichester, UK, Print ISBN: 9781119951506 (Oct. 2012), Online ISBN: 9781118360316 (Jul. 2012), chapter 6, doi: 10.1002/9781118360316.ch6

[9] M. L. Krieg, "A Tutorial on Bayesian Belief Networks", Department of Defense & Technology Organization (DSTO) Electronics and Surveillance Research Laboratory, 2001, available at http://www.dsto.defence.gov.au/publications/2424/DSTO-TN-0403.pdf [last accessed: July 2013]

[10] R. S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, Adaptive Computation and Machine Learning series, MIT Press (Bradford Book), Cambridge, Mass., pp.322, ISBN 0-262-19398-1, 1998

[11] P. Dayan, Ch. JCH Watkins, Reinforcement Learning, Encyclopedia of Cognitive Science, in Press

[12] L. Pack Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement Learning: A Survey," Journal of Artificial Intelligence, Research 4, pp. 237-285, 1996

[13] T. Kohonen, Self-Organizing Maps, Series in Information Sciences, 2nd ed., vol. 30, Springer, Heidelberg, 1997

[14] T. Kohonen, The Self-Organizing map, Elsevier, Neurocomputing 21 1-6 , 1998

[15] Third Generation Partnership Project (3GPP) Technical Specification (TS), "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; Vocabulary for 3GPP Specifications", 3GPP TR 21.905 version 11.3.0 Release 11, 2013-1

[16] Third Generation Partnership Project (3GPP) Technical Specification (TS), "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; Telecommunication management; Integration Reference Point (IRP) Concept and definitions", 3GPP TS 32.150 version 11.2.0 Release 11, 2013-4

[17] Third Generation Partnership Project (3GPP) Technical Specification (TS), "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; Telecommunication management; Integration Reference Point (IRP) overview and usage guide", 3GPP TS 32.103 version 11.2.0 Release 11, 2013-4

[18] Third Generation Partnership Project (3GPP) Technical Specification (TS), "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; Telecommunication management;

Performance Management (PM); Performance measurements; Definitions and template", 3GPP TS 32.404 version 11.0.0 Release 11, 2012-10

[19] Third Generation Partnership Project (3GPP) Technical Specification (TS), "LTE; Telecommunication management; Performance Management (PM); Performance measurements Evolved Universal Terrestrial Radio Access Network (E-UTRAN)", 3GPP TS 32.425 version 11.4.0 Release 11, 2013-1

[20] Third Generation Partnership Project (3GPP) Technical Specification (TS), "LTE; Evolved Universal Terrestrial Radio Access Network (E-UTRAN); S1 Application Protocol (S1AP)", 3GPP TS 36.413 version 11.3.0 Release 11, 2013-04

[21] Third Generation Partnership Project (3GPP) Technical Specification (TS), "LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification", 3GPP TS 36.331 version 11.3.0 Release 11, 2013-04

3 MACHINE LEARNING TECHNIQUES IN COGNITIVE NETWORKS

3.1 Chapter Outline

Rapid evolution of technologies, especially in user terminals, produces vast amount of data, which may refer to information related to a) services/ application, i.e. service data, b) users, i.e. user data or c) networks, i.e. network data. As a result, network operators frequently find it difficult to handle them in terms of adequately processing and exploiting the information that they carry. Moreover, even in cases where data are properly processed and the information they carry is well exploited, the large size of the produced information and their disparity increases the complexity of the system.

On the other hand, cognitive radio systems offer the possibility of processing in an automated way the raw network, service and user data and thus, developing elaborated data in order to facilitate the exploitation of the necessary information from the operators. In other words, cognitive radio systems offer suitable mechanisms for building knowledge and delivering this knowledge in the form of elaborated data and/ or alarms to network operators for better exploitation of the information. Additionally, they are capable of relaxing the complexity of the system through their learning mechanisms. In particular, machine learning mechanisms have proved to be very promising towards both directions, i.e. relaxation of the complexity and increase of data usefulness and thus, have attracted enough researchers.

In general, a machine learning mechanism is fed with raw data and returns patterns or the requested knowledge/ information. For achieving this, the mechanism goes through a process known as training which enables it to build its knowledge. Moreover, learning mechanisms are divided into three basic categories, namely: a) the supervised learning techniques, b) the unsupervised learning techniques and c) the reinforcement learning techniques. These categories are distinguished according to training method they use, i.e., the use of the desired outcome and/ or an award during the training of the algorithm. Specifically, the supervised learning techniques use the desired outcome in order to guide the algorithm what the output should be with respect to the input. The reinforcement learning techniques "award" the system when it comes up with the correct answer during their training. Finally, the unsupervised learning techniques use neither the desired output nor an award during their training. On the contrary, they are able to identify the patterns of the data and exploit them in order to conclude the output.

The chapter describes some representative examples of machine learning techniques and their mathematical models.

The keywords of this chapter are: supervised, unsupervised, machine learning, training and knowledge building.

3.2 Bayesian Statistics

Bayesian statistics are used to estimate future states based on the past input, in case of CRSs where the future behaviour of networks is based on collected measurements. The objective is to estimate the future state using the observations up to the current state. This estimation is modelled as a Probability Density Function (PDF). Bayesian Networks have been proven to be a valuable tool for encoding and learning the probabilistic relationships, as they provide a simple yet effective approach to construct and handle statistical models. A Bayesian network is a graphical model that depicts a set of variables and their probabilistic interdependencies.

3.2.1 Overall cognitive process

In summary, the overall learning process evolves as follows. Observations are collected and fed to the algorithm. Based on these observations, the conditional probabilities, which provide an estimation of how probable it is that a specific under observation parameter will reach a certain value, are updated. The next step is the update of the PDF. The PDF offers a more aggregate estimation regarding the probability to achieve a certain combination of observed parameters given a certain event.

3.2.2 Fundamental elements leading to knowledge

Conditional probabilities. The fundamental elements on which the knowledge can be based, are conditional probabilities that have the form $\Pr\left[V_j = r_{ij}^k \middle| N = i\right]$, where $r_{ij}^k \in$ *R*R_{ij} denotes the k-th reference value for the j-th observed parameter when the event i is considered. These conditional probabilities express the likelihood that the j-th parameter will be equal to the reference value r_{ij}^k, given event i. Table 3.1 depicts the organization of information, for an arbitrary event i∈CN. More specifically, Table 3.1 is known as a Conditional Probability Table (CPT) and serves as a table where all conditional probabilities for a possible event i are gathered. In such table, each row corresponds to one of the observed parameters and each column to one of the reference values.

Probability density function. The following PDF can be defined by:

$$f(\tilde{\cdot} \qquad = i] \cdot \prod_{j=1}^{M} \Pr[V_j = r_{ij}^k \mid N = i] \tag{3.1}$$

where i∈CN, $\tilde{\cdot}$, $r_{ij}^k \in XR_i$ (j= 1,…,M), and k is an integer taking value from 1 to |Rij|.

The sum of the $f(\tilde{\cdot}$ values, overall $\tilde{\cdot}$ and i∈CN, is one. The $\Pr[N = i]$ probabilities show the volume of information existing for event i. The sum of the $\Pr[N = i]$ quantities, over all i∈CN, is 1.

Knowledge. The PDF $f(\tilde{\cdot}$ expresses the knowledge in an aggregate manner on how probable it is that event i will achieve the combination of selected parameters indicated by the vector $\tilde{\cdot}$. Therefore, the $f(\tilde{\cdot}$ contributes to increasing the reliability of the algorithm, since it can take into account the knowledge expressed through the probability associated with the $\tilde{\cdot}$ vector.

3.2.3 Update of the conditional probabilities

This sub-section describes the method for updating the conditional probabilities $\Pr[V_j = r_{ij}^k \mid N = i]$ (which appear in the right end of Equation (3.1), according to approaches suggested in [3][4][5] and in a similar manner to that followed in [2], [6][7][8].

Table 3.1: Organisation of the basic information elements (for arbitrary network i) on which the cognitive mechanisms are based

Parameter	***Reference Value***			
1	r_{i1}^{1}	….	r_{i1}^{k}	….
	$PR[V_1 = r_{i1}^{1} \mid N = i]$	….	$PR[V_1 = r_{i1}^{k} \mid N = i]$	….
….	….	….	….	….
	….	….	….	….
J	r_{ij}^{1}	….	r_{ij}^{k}	….
	$PR[V_1 = r_{ij}^{1} \mid N = i]$	….	$PR[V_1 = r_{ij}^{k} \mid N = i]$	….
….	….	….	….	….
	….	….	….	….
M	r_{iM}^{1}	….	r_{iM}^{k}	….
	$PR[V_1 = r_{iM}^{1} \mid N = i]$	….	$PR[V_1 = r_{iM}^{k} \mid N = i]$	….

It is assumed that observations are collected for each of the events in the candidate events set CN. So, the $\Pr[N = i]$ quantities can be taken equal to the number of collected observations for event i, divided by the total number of observations.

The update of the conditional probabilities $\Pr[V_j = r_{ij}^{k} \mid N = i]$ can take into account the "distance" of measurements from reference values. Let us assume that the most recent observation indicates that event i can achieve a value of V_{ij} regarding the j-th parameter. Let dif_{ij} be the difference between the maximum and the minimum reference value in the set of reference values R_{ij}. Then, for each reference value, $r_{ij}^k \in R_{ij}$, there can be a correction factor, $cor_{ij}^k = 1-(|r_{ij}^k - Vi_j|/dif_{ij})$, where $0 \le cor_{ij}^k \le 1$. A correction value close to one means that the reference value and measured value are close, and thus, the corresponding conditional probability value should be reinforced accordingly. The opposite holds, if cor_{ij}^k is close to zero. The new value of a conditional probability, $\Pr[V_j = r_{ij}^k \mid N = i]$, can be obtained as the product of the value of the old value, the correction factor cor_{ij}^k, and a normalization factor nf_{ij}.

The normalization factor nf_{ij} in this case is used to ensure that the updated values of all conditional probabilities for a certain parameter given a specific event will sum up to one. Moreover, in order to ensure adaptability to new conditions, the conditional probabilities are prohibited from falling below a certain threshold, p_{min}. Implicitly, this also means that the conditional probabilities are not allowed to exceed a certain threshold, $p_{\max} = 1-(|R_{ij}|-1)\cdot p_{\min}$. In summary, the update strategy includes: (i) collection of measurements; (ii) computation of the correction factors, of the normalization factor, and of the new values of the conditional probabilities; (iii) the L probabilities that may fall below p_{min} are set equal to the threshold; (iv) the remaining probabilities that have not fallen below the p_{min} threshold are equally reduced so as to sum to $(1-L\cdot p_{\min})$. After the update of the conditional probabilities values, the update of the PDF follows. This is realised through the use of Equation (3.1).

3.3 Supervised Neural Networks (NNs)

NN is an artificial mechanism which attempts to adopt the way human neurons interwork in human bodies. As such, they consist of neurons which are interconnected to each other in a common programming structure. This kind of mechanisms prove to successfully address narrowly defined problems such as problems related to pattern (speech/image) recognition, time-series prediction and modelling, function approximation, classification, adaptive control and other areas.

To begin with, the structure of the neurons is divided into three parts: a) the input layer, b) the output layer, and c) the intermediate part that may consist of one or more hidden layers (upper part of Figure 3.1). In particular, neurons of the input layer are responsible for receiving data from the external environment of the NN. The output layer's role involves the transmition of the results of the NN towards the external environment/ user. And the last part is the one that basically processes the data.

Based on their topology, NNs can be divided into two basic (non exhaustive) types: a) the feed-forward NNs and b) the recurrent NNs. In feed-forward NNs, data enters the NN through the input layer, and passes from layer to layer until they reach the output layer. Some classical examples of this type are the Perceptron [9] and Adaline [10]. The recurrent NNs are further equipped with connections that originate from the output of the neuron and feed neurons of the same or previous layers giving a sense of history and awareness of events from previous time steps. Typical examples of this type have been presented by Elman [11] and Hopfield [12].

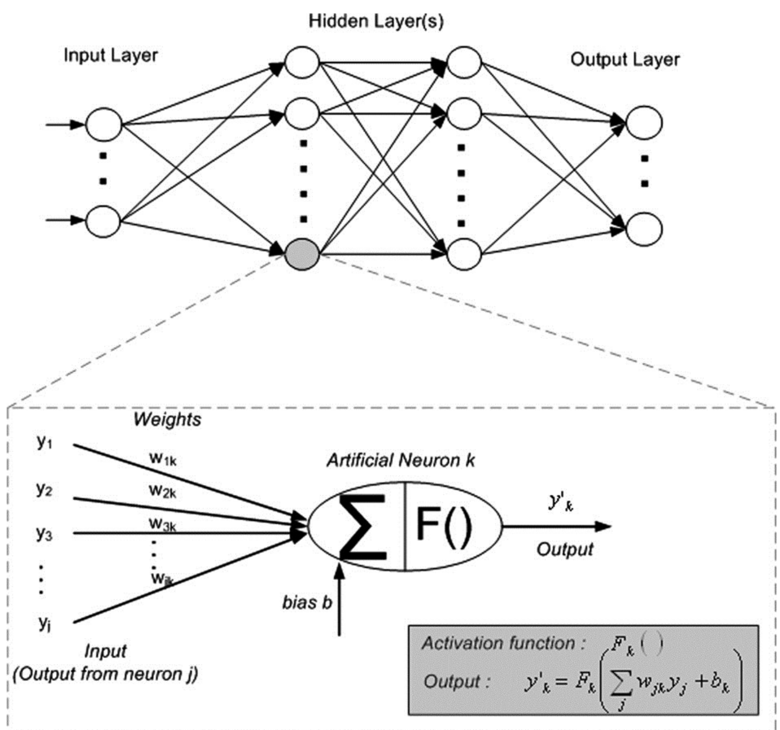


Figure 3.1: Typical neural network structure [1]

For processing data, the activation function F_k in (3.2) is used so as to update the output of neuron k, y'_k (see also the lower part of Figure 3.1). In this function, weight w_{jk} stands for the weight carried by the connection between neurons j and k in terms of the effect that the signal of the former has on the latter. Moreover, y_k designates the output of the neuron, known as state of activation, and b_k represents inputs of neuron k by external sources a.k.a bias offset.

$$y'_k = F_k(\sum_j w_{jk} y_j + b_k) \tag{3.2}$$

Focusing on the inputs coming from neighbours of different layers $w_{jk}y_j$, some sort of threshold functions can also be used. Some types of such functions are summarized hereafter in Equations (3.3)-(3.7):

- sign function $$F_k(w_{jk}y_j) = \begin{Bmatrix} -1 & if & x<0 \\ 0 & if & x=0 \\ 1 & if & x>0 \end{Bmatrix} \tag{3.3}$$
- linear function $$F_k(w_{jk}y_j) = aw_{jk}y_j + b \tag{3.4}$$
- logistic-sigmoid transfer function (logsig) $$F_k(w_{jk}y_j) = \frac{1}{1+e^{-w_{jk}y_j}} \tag{3.5}$$
- hyperbolic tangent sigmoid transfer function (tansig) $$F_k(w_{jk}y_j) = \frac{2}{1+e^{-2w_{jk}y_j}} - 1 \tag{3.6}$$
- linear transfer function (purelin) $$F_k(w_{jk}y_j) = w_{jk}y_j \tag{3.7}$$

For building the desired knowledge, neural networks need to be trained, i.e., need to adjust the weights w_{jk} of the connections between all possible combinations of neurons (j,k), so as to produce the desired output when a particular input is considered. Towards this direction, the selected NN is fed with an input and is left to calculate its output. As soon as the output is produced, it is compared to the desired output. The error between the two values is then split in error values (one per connection) which in the sequel are back propagated from the output layer to the neurons of the hidden layers. Thereinafter, neurons proceed to the respective changes in order to minimize the error between the produced and the desired output. The training (or learning) procedure ends when the weights on the connections between neurons are properly adjusted so as to encode the actual knowledge of the NN, leaving the NN capable of being used for the purpose that is initially set up for. It is worth mentioning at this point that caution is needed when training the NN so that the latter not to be overtrained, i.e., not to learn features of the pattern that apply only to the training data set and prevent the NN from applying successfully the pattern to other data. By the term "training data sets", we refer to those sets of data that are used for the training in the first place. The above mentioned undesired phenomenon would eventually result in a NN that would not be able to generalize well.

3.4 Self-Organizing Maps (SOMs): an unsupervised neural network

SOM is an unsupervised learning technique that is based on neural networks and was introduced by T. Kohonen in [13], while a short overview of its theory foundation can also be found in [14] and [15].

In particular, SOM has two very interesting attributes that make it very attractive for data mining and classification problems, i.e., the ability of depicting multi-dimensional data in 2D maps and the ability of depicting similar data close to each other. Due to these two attributes, SOM has widely been used for many different applications in science fields. More specifically, authors in [16] exploit the classification provided by SOM for distinguishing samples of illicit drugs and categorizing them among six specific types (methyl ephedrine hydrochloride, cocaine hydrochloride, ephedrine hydrochloride, methadone hydrochloride, pseudo ephedrine hydrochloride and narceine hydrochloride), while [17] presents an application of the technique for analysing chemicals. Researches using this technique have also focused on document collections [18], speech recognition [19], identification of a cancer cell gene [20], hematopoietic differentiation [21] and manipulation of security threats [22]. Further initiatives and applications of this technique in many science fields can also be found in [23].

In general, as also imposed by its name, SOM is a 2D map that comprises of rectangular or hexagonal cells ordered on a regular grid. For the representation of the multi-dimensional data on this grid, a training process is required. In particular, the data is inserted in the training process as data samples, each weight of which refers to another dimension of the data. It is worth mentioning at this point that the data samples may enter the process in two different ways, i.e., one by one or in parallel, resulting in two training algorithms, i.e., the sequential and the batch training algorithm respectively. The difference between the two algorithms originates from the way of the entrance of the data samples and extends to the sequential or parallel processing of the data samples during the whole training process making the batch training algorithm faster.

Furthermore, the training of the map involves the comparison of each vector (data sample) to the vector of each cell. The cell whose vector is the most similar to the data sample is called Best Matching Unit (BMU) and its vector is finally adjusted so as to become more similar to the data sample. According to the technique, apart from the vector of BMU, the vectors of a neighbourhood around BMU may also be adjusted according to the data sample. The adjustment or not of the neighbourhood, the way of the adjustment and the respective neighbourhood are set by the user through a function known as neighbourhood function. Figure 3.2 depicts the training process which results in the ordered SOM, where the more similar the data of the cells, the closest the cells to each other. In this term, the created map represents the similarity of the data and their classification. Further details, and the respective mathematical foundation of both the sequential and the batch training algorithms, can be found in the following paragraphs of this section.

To sum up, the steps of the training of the map are depicted in Figure 3.3 and are as follows: (1) the map is initialized, i.e., each of its elements/cells is represented by a vector that is comprised of as many components as the dimensions of the data; (2) each data sample is also expressed as a vector with weights that are equal to the values of the dimensions of the data sample and is mapped on the cell whose vector is closest to it when using Euclidean distance; (3) the most important part is that the vectors of the data that are inserted into the training process of the map, also influence the weights of the map (cells) vectors as to adjust them closer to their weights; (4) the end of the training process finds the map complemented with the multi-dimensional data and split into clusters since the distance between the data samples is now represented by the distance between the cells of the map.

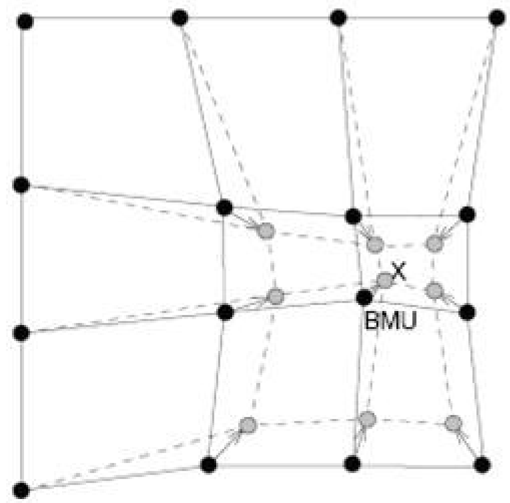


Figure 3.2: The inserted data sample x affects its BMU and its neighbourhood. The solid and dash-doted lines correspond to the situation before and after the input of the data sample [15]

After the first introduction of SOM from Kohonen, there have been different approaches attempting to enhance the basic algorithm by enforcing its flexibility. Some such examples are the Growing SOM [24] which enables the map to adjust its size according to its need for better organizing the data samples, the Parameterless SOM [25] that provides flexibility in terms of the neighbourhood around the BMU that will be affected by eliminating it from the predefined by the user variables, and the Hierarchical SOM [26] that grows in interacting layers and hybrids of the above described [27][28].

3.4.1 Sequential Training Algorithm

In the sequential training algorithm, every data sample enters the process by its own (sequentially) making the algorithm iterative. Each iteration t starts when a data sample x is inserted, and ends when the training of the map that is caused by the data sample x has finished. As already mentioned above, the training of the map involves the insertion of the data sample x, the identification of its BMU c and the update of the vectors of the BMU m_c and of the neighbouring cells m_i. For achieving this, the distance of each data sample x from each vector of the SOM m_i is calculated. The minimum distance refers to the BMU c of the data sample x, i.e., the cell whose vector m_c is closest to the data sample x. The corresponding equation of the above described process is the following (Equation (3.8)):

$$\| x(t) - m_c(t) \| \leq \| x(t) - m_i(t) \| \tag{3.8}$$

where ||.|| stands for the Euclidean Distance.

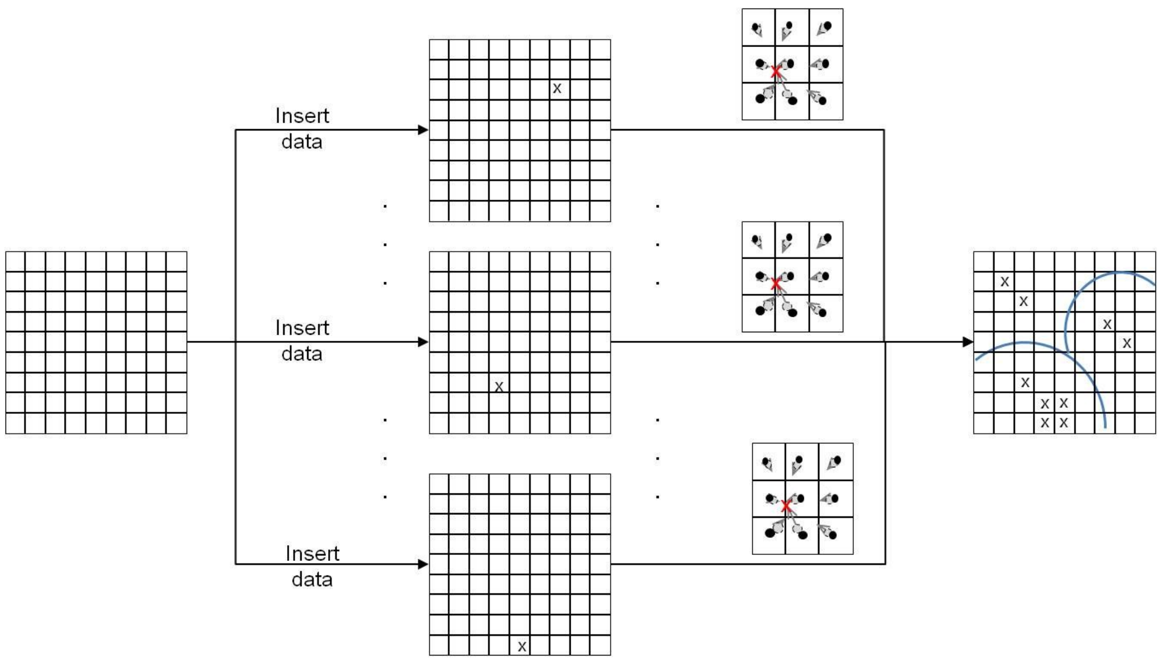


Figure 3.3: Overview of the SOM technique

The identification of the BMU c is followed by the adjustment of its vector (and often of the vectors of neighbouring cells) so as the weights of m_c to become more similar to the weights of x. In particular, m_c and the vectors of the neighbouring cell m_i are updated according to Equation (3.9):

$$m_i(t+1) = m_i(t) + a(t) h_{ci}(t)[x(t) - m_i(t)] \tag{3.9}$$

where α(t) is the learning rate factor which is responsible to tell how much each cell will be influenced by the specific data sample x(t), and $h_{ci}(t)$ is the neighbourhood function.

The four functions that may be used for calculating $h_{ci}(t)$ are functions (3.10) - (3.13):

- Bubble: $h_{ci}(t) = l(\sigma_t - d_{ci})$ (3.10)

- Gaussian: $h_{ci}(t) = e^{-d_{ci}^2/2\sigma_t^2}$ (3.11)

- Cutgauss: $h_{ci}(t) = e^{-d_{ci}^2/2\sigma_t^2} l(\sigma_t - d_{ci})$ (3.12)

- Ep: $h_{ci}(t) = \max\{0, 1-(\sigma_t - d_{ci})^2\}$ (3.13)

where σ_t is the radius of the neighbourhood, i.e., corresponds to the number of the cells that will be influenced by the data sample x, $d_{ci} = \| r_c - r_i \|$ is the distance between the cells m_c and m_i, and l(x) is the step of the function: $l(x) = 0$ if $x < 0$ and $l(x) = 1$ if $x \geq 0$.

Accordingly, factor α(t) can be calculated with respect to the next three different functions (3.14) - (3.16):

- Linear function: $a(t) = a_0(1 - t/T)$ (3.14)

- Power function: $a(t) = a_0(0.005/a_0)^{t/T}$ (3.15)

- Inv function: $a(t) = a_0/(1 + 100\,t/T)$ (3.16)

where T is a constant variable, called training length and a_0 is also a constant variable, known as initial learning rate.

3.4.2 Batch Training Algorithm

Batch training algorithm is also iterative. However, when this algorithm is applied, all data samples are inserted and presented to the map simultaneously before any adjustment is made. In each training step t, data samples are mapped according to the Voronoi regions of the map weight vectors, thus they are mapped to the cells whose weight vectors are closest to them. The new vector of each cell m_i is given by the following relation (3.17):

$$m_i(t+1) = \frac{\sum_{j=1}^{n} h_{ic}(t) x_j}{\sum_{j=1}^{n} h_{ic}(t)} \tag{3.17}$$

Where $c = \arg\min_k \{\| x_j - m_k \|\}$ denotes the BMU of data sample x_j and $h_{ic}(t)$ is the neighbourhood function. As well as in the sequential training algorithm, it can also be computed by formulas (3.10) - (3.13).

An alternative approach can also be followed. In particular, the new vectors of the cells can be calculated by Equation (3.18) after having calculated the sum of the vectors of each Voronoi region of the map by Equation (3.19).

$$m_i(t+1) = \frac{\sum_{j=1}^{m} h_{ij}(t) s_j(t)}{\sum_{j=1}^{m} n_{v_i} h_{ij}(t)} \tag{3.18}$$

$$s_i(t) = \sum_{j=1}^{n_{Vi}} x_j \tag{3.19}$$

In the above Equations (3.18) - (3.19), m denotes the number of cells of the map and n_{Vi} is the number of the mapped to the cell i data samples.

3.5 Reinforcement Learning

Reinforcement Learning (RL) in general mimics the way that animals learn how to optimize their behaviours when punishments and rewards of their actions apply. In particular, RL algorithms follow the next three phases:

a. *Observation of their environment and identification of the current state.* During this phase, the system observes its environment and decides the type of information that is needed for describing better its current state with respect to the optimization problem. As soon as the type of the needed information has been identified, the system collects the specific data that describes the current state, in terms of context and circumstances.

b. *The system acts.* In this phase, a decision has been reached and executed making the system to move towards its new state. In cases where the system is already familiar with what should be done, i.e., which action will be rewarded and which will not, the action is selected in order to lead to a reward. On the contrary, if the system is still "young", the actions are selected arbitrarily. For training the system well, different actions should be performed under the same conditions (state) so as to have as much feedback as possible for their correctness or not.

c. *Evaluation of the action.* At this point the system receives an evaluation of the taken action in terms of an immediate numeric payoff. This payoff stands for punishment if it is a negative number or for reward if the number is positive. It is important here to clarify that this payoff is subjective since it depends on the experience and the prior knowledge of the latter.

The final target of the RL is to maximize either the long term or the average sum of these payoffs.

Moving from theory to mathematical formulation, there are two mathematical models leading to two different, yet similar, learning techniques. The two respective learning techniques are known as: a) Actor critic learning and b) Q-learning. Although both techniques in practice are found to work well, the circumstances and a solution which they converge to, i.e., the action with the better reward, are known only for the second one. Thus, here we focus on this technique, i.e., Q-learning. During the learning process, the system at time t identifies its current state s(t) and decides its action a(t). As a result of this action, the system receives the respective payoff r(s(t),a(t)). Moreover, the system moves to the next state through a transition distribution $P_{xy}(a)$ which reveals the probability of the system to move from state x to state y when action a is applied. Given this context, the target of RL is transformed to the maximization of (3.20).

$$Q(s(t),a(t)) = \left\langle \sum_{t=0}^{\infty} \gamma^t r(s(t),a(t)) \right\rangle_{s,r} \tag{3.20}$$

where Q(s(t),a(t)) gives the quality of the combination of state s(t) with the action a(t), symbol $\langle\ \rangle_{s,r}$ refers to the average value and 0<γ<1 stands for discount factor. The latter represents the weight of the payoff and is closely related to the time passed from the payoff, i.e., the larger γ designates that the more distant payoffs are more important.

During this process, Q(s(t),a(t)) keeps being updated though (3.21) until it reaches its optimal value, i.e., until it reaches its maximum value. In this function, ε denotes the learning rate of the system.

$$Q(s(t),a(t)) \to Q(s(t),a(t)) + \varepsilon[r(t) + \gamma \max_{b} Q(s(t+1),b) - Q(s(t),a(t))] \qquad (3.21)$$

Finally, the most appropriate action, i.e., the policy p that dictates the next action of the system, can be calculated through Equation (3.22)

$$p(s) = \arg\max_{a}\{Q(s,a)\} \qquad (3.22)$$

Further information and details with respect to this technique can also be found in [29], [30] and [31].

3.6 Chapter References

[1] K. Tsagkaris, A. Katidiotis, P. Demestichas, "Neural Network-based Learning schemes for Cognitive Radio systems," Computer Communications, vol. 31, Issue 14, pp. 3394-3404, 2008

[2] O. Orcay, B. Ustundag, "Pattern recognition in cognitive communication," 23rd International Symposium on Computer and Information Sciences, pp.1-6, 27-29 Oct. 2008

[3] Stuart J. Russell, Peter Norvig, Artificial Intelligence: A Modern Approach, Prentice-Hall, New Jersey, 2002

[4] R. E. Neapolitan, Learning Bayesian networks, Prentice Hall (series in artificial intelligence), 2002

[5] F. Jensen, Bayesian Networks and Decision Graphs, Springer-Verlag, New York, 2001

[6] A. Koutsorodi, E. Adamopoulou, K. Demestichas, M. Theologou, "Service configuration and user profiling in 4G terminals," Wireless Personal Communications, vol. 43, no. 4, pp.1303-1321, Dec. 2007

[7] K. Demestichas, A. Koutsorodi, E. Adamopoulou, M. Theologou, "Modelling user preferences and configuring services in B3G devices," Wireless Networks, vol. 14, no. 5, pp. 699-713, Oct. 2008

[8] Y. Kritikou, V. Stavroulaki, P. Demestichas, "A theoretical approach for learning user behaviour and preferences as part of the management of cognitive wireless B3G infrastructures," in Proc. 11th Symposium on Wireless Personal Multimedia Communications 2008 (WPMC 2008), Lapland, Finland, September 2008

[9] Frank Rosenblatt, "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain," Cornell Aeronautical Laboratory, Psychological Review, vol. 65, no. 6, pp. 386–408, 1958

[10] B. Widrow, S.D. Sterns, Adaptive Signal Processing, Prentice-Hall, New York, 1985

[11] J. L. Elman, Finding Structure in Time, Cognitive Science 14: 179-211, 1990

[12] J.J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," Proc. Natl. Acad. Sci. USA, 79 (1982) 2554–2558

[13] T. Kohonen, Self-Organizing Maps, Series in Information Sciences, 2nd ed., vol. 30, Springer, Heidelberg, 1997

[14] T. Kohonen, The Self-Organizing map, Elsevier, Neurocomputing 21 1-6 , 1998

[15] J. Vesanto, J. Himberg, E. Alhoniemi, J. Parhankangas, SOM Toolbox for Matlab 5. Technical Report A57, Neural Networks Research Centre, Helsinki University of Technology, Helsinki, Finland, 2000

[16] M. Liang, J. Shen, G. Wang, "Identification of illicit drugs by using SOM neural networks", Journal Of Physics D: Applied Physics, 41, 2008

[17] H. Tokutaka, K. Yoshihara, K. Fujimura, K. Obu-Cann, K. Iwamoto, "Application of self-organizing maps to chemical analysis", Elsevier, Applied Surface Science, 144-145 59-63, 1999

[18] S. Kaski, T Hankela., K. Lagus, T. Kohonen, "WEBSOM – Self-organizing maps of document collections", Elsevier, Neurocomputing 21 101-117, 1998

[19] T. Kohonen, P. Somervuo, "Self-Organizing Maps of Symbol Strings with Application to Speech Recognition", in Proceedings of Workshop on Self-Organizing Maps (WSOM'97), Espoo, Finland, pp. 2-7, June 4-6, 1997

[20] Y. Matsuura, M. Tuoya, Seno, H. Tokutaka, M. Ohkita, "The Identification of a Cancer Cell Gene by using SOM", 16th International Conference on Genome Informatics, Yokohama Pacifico, Japan, December 19-21, 2005.

[21] P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. S. Lander, T. R. Golub, "Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation", Proc. Natl. Acad. Sci. USA, vol. 96, pp. 2907-2912, 1999

[22] T. Clancy, A. Khawar, "Security Threats to Signal Classifiers Using Self-Organizing Maps," CrownCom, June 2009

[23] Laboratory of Computer and Information Science, Adaptive Informatics Research Centre, http://www.cis.hut.fi/research/refs/

[24] T. Villmann and H.-U. Bauer, "Applications of the growing self-organizing map," Neurocomputing, vol. 21, no. 1–3, pp. 91–100, 1998

[25] E. Berglund and J. Sitte, "The parameterless self-organizing map algorithm," IEEE Transactions on Neural Networks, vol. 17, no. 2, pp. 305–316, 2006

[26] S. T. Sarasamma, Q. A. Zhu, J. Huff, and Hierarchical Kohonenen, "Net for Anomaly Detection in Network Security," IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics, vol. 35, no. 2, April 2005

[27] M. Dittenbach, D. Merkl, and A. Rauber, "Growing hierarchical self-organizing map," in Proceedings of the International Joint Conference on Neural Networks (IJCNN '00), vol. 6, pp. 15–19, July 2000

[28] T. Kuremoto, T. Komoto, K. Kobayashi, and M. Obayashi, "Parameterless-Growing-SOM and Its Application to a Voice Instruction Learning System, Research Article," Journal of Robotics, vol. 2010, Article ID 307293, 9 pages, doi:10.1155/2010/307293

[29] R. S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, Adaptive Computation and Machine Learning series, MIT Press (Bradford Book), Cambridge, Mass., pp.322, ISBN 0-262-19398-1, 1998

[30] P. Dayan, Ch. JCH Watkins, Reinforcement Learning, Encyclopedia of Cognitive Science, in Press

[31] L. Pack Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement Learning: A Survey," Journal of Artificial Intelligence, Research 4, pp. 237-285, 1996

4 KNOWLEDGE BUILDING MECHANISMS FOR ESTIMATING NETWORK CAPABILITIES

4.1 Chapter Outline

The aim of this chapter is to present a knowledge building mechanism that estimates network capabilities and to provide comparative studies of such mechanisms which however are based on different machine learning techniques.

The term "network capabilities" refers to what the network is capable of, i.e., the main features of a network such as the QoS, its range, its location, its type (GSM, UMTS), etc. In this study, the term refers explicitly to the QoS that the network may offer. Consequently, QoS may also refer to more than one parameter, such as the bitrate, the jitter, the delay, the bit error rate and the throughput of the network. In this case, QoS is mentioned in terms of achievable bitrate. Summarizing, the scope of this case is to estimate network capabilities in terms of QoS, expressed in bitrate, based on current network measurements and context. It is worth mentioning at this point that by network measurements, measurements that refer to parameters holding information related to the network identity, its Radio Access Technology (RAT), its configuration, its Received Signal Strength Identifier (RSSI) and its traffic, in terms of packets or Bytes, are considered. Moreover, context refers to those parameters that hold information such as time, location and the environmental conditions.

The keywords of this chapter are: CRS, Cognition Cycle, Unsupervised Learning, SOMs

4.2 Problem Statement

The focus of this study is placed on using SOMs in order to assist CRSs to choose among the different candidate radio configurations to operate with, by taking into account the predictions of the bitrate that can be achieved. Actually, this study comprises a significant extension, in terms of scenarios and comparative analysis, of the work that was presented in [5]. A summary of the results in [5] is given in sub-section 4.4.1. This study extends this work by introducing scenarios that consider pre-processing of the data (sections 4.4.2 and 4.4.3) used for training. Moreover, the results of the new extended scenarios are compared to those of [5] in sub-section 4.4.4. Finally, a comparison between the results of this research and the respective results of a similar research that exploits supervised NN-based learning techniques is given in section 4.5 as well.

This study examines the possibility of connecting parameters observed while tuning at a specific configuration, such as RSSI, errors (input and output), packets (received and sent) and bytes (received and sent) with one QoS metric i.e. the achievable data rate, in order to predict it. This is done by using the unsupervised training technique provided by SOM. However, SOM is a technique for the representation and classification of multidimensional data into 2D maps. So how could it be useful in this case?

According to SOM basics (see also section 3.4), this technique is able to categorize multidimensional data samples, as long as their vectors have the same dimensions (number of used variables per data sample) and the same type of variables (used parameters for creating the data samples), and to organize them according to their similarity. The analysis of the experimentation below (sub-section 4.3) presents how a different case, i.e. a case where the number of used parameters in the data sample differs, can be treated. More precisely, the algorithm has the ability to ignore variables of the data sample that do not have any value, i.e., missing values, during the calculation of the distances.

Moreover, it is worth clarifying here that the proposed method offers flexibility towards differentiating the number and the type of the parameters without demanding its redesign. This can be done by easily changing the number and/ or the type of the parameters that comprise the data samples which train the SOM at the first place (more information regarding this is also available at the next section, i.e. sub-section 4.3 – experimentation setup). In any case, the parameters that will be used for estimating the bit rate obtained with the specific configuration have to be the same with the parameters that are used during the training of the map.

Based on the attribute of SOM to classify multidimensional data into 2D maps, the method was set to formulate a number of different groups (clusters). Specializing the above in this case study the role of the multidimensional data is played by the data samples whose variables may be the RSSI, the input and output errors, the received and sent packets and the received and sent bytes. As a result, data samples, whose values of the participating variables are similar, get organized together. The data samples/ nodes of each cluster are selected according to the respective achieved raw data rate, given the values of the observed variables during the configuration under question. Note that the measured raw data rate is selected to be inserted into the system for distinguishing the data samples (labelling), but not for participating in the formulation of the map. Accordingly, each cluster contains data with the same measured raw bitrate. The proposal of this study is based on the fact that if there is a new data entry, which consists of the same variables and if the SOM theory is applied, then it is depicted on the map as part of an existing cluster. Accordingly, this data sample is expected to be similar to the other participants of the cluster and to exhibit the same raw bitrate. Thus, in this case study, the unknown raw data rate of a new data sample is expected to be equal to the bitrate of the data samples used for creating the corresponding cluster in the first place. With respect to this methodology, the achieved bitrate of a configuration can be predicted based on the values of RSSI, errors (input and output), packets (received and sent) and bytes (received and sent).

Last but not least, a major contribution of this work is the actual implementation, testing and performance evaluation of the proposed method and its respective comparison with the results of other learning-based methods. The remaining sub-sections are completely devoted to the description of the setup of the experimentation part, of the conducted scenarios and test cases and eventually of the obtained results.

4.3 Experimentation setup

In order to validate the proposed learning method, commercial off-the-shelf hardware and software products were used and extended. More specifically, the data used for the test cases have been obtained from measurements that took place in a real working environment within our university premises. Particularly, a laptop equipped with an Intel 3945ABG Wireless card has been used for measuring the maximum achievable raw transmission data rate (bitrate), the link quality and the signal strength in user predefined time intervals. The laptop has been setup with a Debian OS running on a

2.6.18 kernel and using the ipw3945 driver. The wireless access point (AP) used was a Linksys Wireless-G broadband router (model WRT54GS) which was able to operate in both IEEE 802.11 b/g standard modes. This actually comprised the radio configuration (it can be seen as one single configuration given that the operating carrier frequency is the same i.e. 2.4GHz in both modes), the capabilities of which need to be discovered-evaluated. The data collection lasted for 7 days and the applications used during that period included peer-to-peer (P2P) file sharing, web browsing and file transfer protocol (ftp).

To begin with, the obtained measurements were used to create different data files. Each file, the format of which is depicted in Figure 4.1, was used as input for the training of the SOM and comprises different combinations of the considered parameters. In particular, each column, apart from the last one, refers to a different observed parameter/ variable of the data sample, while each row corresponds to one different data sample (see Figure 4.1). Finally, the last column of the data file is the measured value of the corresponding bitrate. The latter was used during the experiments for two reasons: a) for distinguishing and organizing into groups the data samples which have the same bitrate and b) for evaluating the technique by comparing the predictions of the method with the real/ measured values.

In the sequel, the created data file(s) and SOM toolbox v.2 of MATLAB [1] were used to train the SOM. The training of SOM demands first of all the insertion of the data, i.e. the loading of the corresponding data file with respect to each test case. Secondly, it is needed to define variables, which concern the topology of the SOM. These variables are: a) the map size, b) the lattice (local lattice structure) and c) the global map shape. The map size refers to the number of the neurons of the SOM. According to Céréghino in [6], "the number of output neurons in a SOM can be selected using the heuristic rule suggested by Vesanto et al. in [1], and applied in [7] in a study of diatom communities: the optimal number of map units is close to $5 \cdot \sqrt{n}$, where n is the number of training samples (sample vectors). In this case, the two largest eigenvalues of the training data are first calculated, then the ratio between side lengths of the map grid is set to the ratio between the two maximum eigenvalues. The actual side lengths are finally set so that their product is close to the number of map units determined according to Vesanto et al.'s rule".

```
5
#n  RSSI   IPKTS  OPKTS  IBYTES  OBYTES
     -69    886   1680   31206   845424  48
     -71    812   1540   28600   774972  36
     -72    860   1680   30348   845424  36
     -73    820   1540   28864   774972  36
     -72    804   1536   28336   772796  48
     -73    756   1404   26588   706696  48
     -73    608   1120   21376   563616  36
     -71    680   1330   23976   669294  48
     -69    948   1821   33416   915876  54
     -70    794   1540   28006   774972  54
     -71    922   1686   32454   845774  54
     -72    712   1394   25076   704170  36
     -72    848   1609   29848   810198  36
     -74    770   1471   27154   739746  36
     -74    698   1262   24510   634134  36
     -78    514    994   18110   493626  24
     -76    558    994   19562   493626  48
     -75    566    994   19826   493626  48
     -76    546    994   19166   493626  36
```

Figure 4.1: Matlab Data File: Each line is a data sample and each column is a different parameter of the configuration. The last column refers to the bitrate which was used as label.

The lattice structure refers to the shape that each neuron has on the SOM and can be "hexagonal" or "rectangular" [1]. Finally, the shape refers to the global map shape and may be "sheet", "cylinder" or "toroid" [1]. The above process, during which the topology of SOM is set, is called initialization. Having the initialization being completed, and after the selection of the training algorithm (batch or sequential) and the values of its respective parameters, the training of the SOM can take place. The parameters of the batch and the sequential training algorithms and their possible values are depicted in Table 4.1 and Table 4.2, respectively. As it is observed in the tables, both training algorithms are divided in two phases: a) the rough phase where the initial learning rate and the neighbourhood radius are relatively large and b) the fine-tuning phase where the initial learning rate and the neighbourhood radius are small from the beginning. As a result, during the first phase the map is approximately shaped while during the second one it is fine-tuned.

Table 4.1: Parameters of batch training algorithm

	Neighbourhood function	h_{ij}	Bubble / Gaussian / Cutgauss / Ep
Rough phase / Fine-tuning Phase	**Initial radius**	$\sigma(t)$	positive integer number or zero, radius of neighbourhood during the first iteration of the phase expressed in cells
	Final radius	$\sigma(t)$	positive integer number or zero, radius of neighbourhood during the last iteration phase expressed in cells
	Training length	T	positive integer number, number of iterations of the algorithm expressed in epochs

Table 4.2: Parameters of sequential training algorithm

	Neighbourhood Function	h_{ci}	Bubble / Gaussian / Cutgauss / Ep
Rough phase / Fine-tuning Phase	**Length type (type of measurement of training length)**		Epochs / Samples
	Learning function	$\alpha(t)$	Linear / Power / Inv
	Initial radius	$\sigma(t)$	positive integer number or zero, radius of neighbourhood at the beginning of the phase, expressed in cells
	Final radius	$\sigma(t)$	positive integer number or zero, radius of neighbourhood at the end of the phase, expressed in cells
	Training length	T	positive integer number, number of iterations of the algorithm (in case of epochs) or number of data samples (in case of samples)
	Initial alpha (initial learning rate)	a_0	Real number

It must be also noted here that the MATLAB SOM toolbox uses slightly modified versions for both the sequential and batch training algorithms presented in sub-section 3.4 [1]. In particular and considering the sequential case first, the computations in equations (3.8) and (3.9) above are slightly modified due to the fact that there may be some missing values of the variables of a data sample or the selected mask may dictate something different. In particular, the distance calculation in equation (3.8) transforms into the next equation:

$$\|x-m\|^2 = \sum_{k \in K} w_k \left(x_k - m_k\right)^2 \tag{4.1}$$

where *K* denotes the set of known (not missing) variables of sample vector x, x_k and m_k are k-th components of the sample and weight vectors w_k is the k-th mask value.

Moreover, equation (4.2) transforms into the next equation:

$$m_i(t+1) = m_i(t) + a\left(t\right)h_{ci}(t)[x(t) - m_i(t)] \tag{4.2}$$

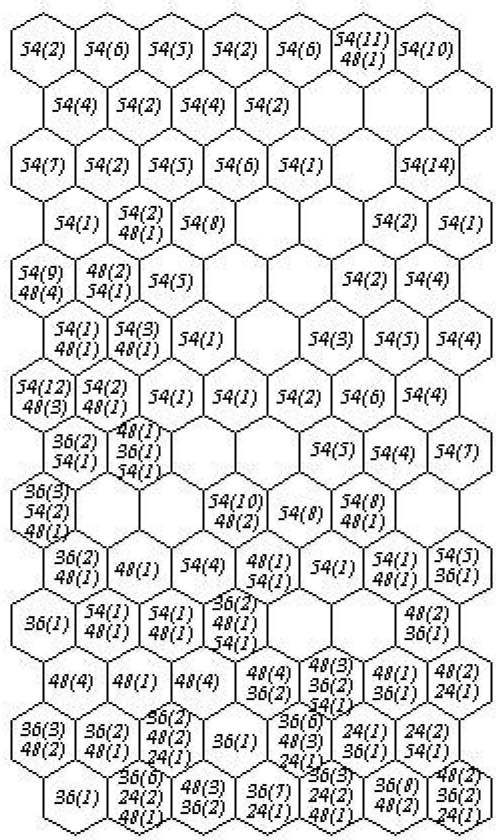


Figure 4.2: Labelled SOM when using the FREQ version

Considering the batch training algorithm, the training is executed using the alternative way i.e., the calculation of the new weights of the vectors follow the calculation of the sum of the vectors in each Voronoi region of the map. Moreover, as well as in the case of sequential training algorithm, the missing values are ignored during the distance calculation.

Furthermore, an essential step prior to training the SOM was the decision upon the normalization of the data. Three different scenarios were examined and test cases were setup according to criteria related to normalization. Specifically, the first scenario referred to data that had not been normalized; the second one referred to data that had been normalized to [0, 1] and the last scenario referred to data whose variance had been normalized to [0, 1]. So, after the normalization (or not) of the data with respect to the scenario, the latter were used for training a SOM map.

For facilitating the analysis, SOM toolbox offers the ability to use labels for distinguishing the category of the data samples. In this case, the category of the data samples was set to designate the bitrate that was related to the data sample (four predefined values, i.e., 24, 36, 48 and 54 Mbps). Thus, each label corresponded to the measured value of the bitrate for the mapped data sample. However, the fact that more than one data samples may have the same BMU (m_c) results in the fact that each cell of the map may have more than one labels appearing more than once. As SOM toolbox offers enough different ways for labelling the map, three of them were used ending up with three different versions of the method / algorithm. The first way (VOTE) is to put on each cell only the most frequently appearing label, the second one (ADD1) is to put all labels while the third one (FREQ) is to put all labels, like in case of ADD1, but in descending order with respect to their appearance frequency and followed by the number of appearances.

At this point, the output of the algorithm is a labelled SOM map, like the one in Figure 4.2, where hexagonal cells are used. This map should not be confused with the used wireless network topology. On the contrary, it could be part of the information carried in the wireless AP of the used topology. Based on this map/ information the program may represent a new data sample on the map but cannot predict its bitrate. In order to train the program how to predict the bitrate of a data sample, the above described visualization needed to be transformed into mathematical functions. The transformation was realized as follows: according to the ability of SOM technique to depict similar data in adjacent cells, cells which were marked with the same bitrate were expected to form a cluster. The centre of each cluster was calculated by the equations

$$x = \sum_{i}^{n} w_i x_i / n \tag{4.3}$$

and

$$y = \sum_{i}^{n} w_i y_i / n \tag{4.4}$$

where n is the number of cells which belong to the cluster, x_i and y_i are the co-ordinates of the cell i and w_i is the weight with which the cell i participates into the calculation. In the first two versions (VOTE, ADD1) w_i is always set equal to 1 while in the last version (FREQ), w_i is calculated by function 4.5:

$$w_i = k/r \tag{4.5}$$

where k is the number of instances with the specific bitrate in the cell i and r is the total number of instances of cell i, i.e., the sum of the instances of all bitrates of the cell. It must be noted here that this method is limited to data that their raw bitrate varies between 24, 36, 48 and 54 Mbps, respectively, and thus the created clusters that were formed were only four, one for each of those possible bitrates.

In order to define the bitrate of a data sample, the cluster in which the BMU of the data sample belonged to was required. The BMU was set to belong to the cluster with the closest centre when using the Euclidean distance. As a result, the bitrate of the data sample will be the one that represents the cluster. At this point, each data sample has a prediction of its corresponding bitrate. This prediction is to be compared to the real measured value of the bitrate, which is part (last column) of the data file used to insert the data samples into the system, for evaluating the method and reaching conclusions. Results from executing various test cases during this evaluation showed a satisfying ability of the method in correctly predicting the bitrate of the data samples as will be revealed in sub-section 4.4.

4.4 Test Cases and Results

As already stated, three different scenarios were set up differing in the type of normalization used. Moreover, in each scenario, a number of test cases that corresponded to variations of input parameters of the proposed method were also set up in order to validate the latter. In particular, the focus was placed on exploring the following aspects: a) which is the best choice between the three labelling options (VOTE, ADD1 and FREQ), i.e., the optimal way of calculating the centres of the clusters, b) what variables of the data samples should be used, c) how many data samples are needed for the training phase and d) what the training algorithm and the values of its parameters should be. In addition, the metric used for evaluation and comparison reasons was the percentage of the data samples the bitrate of which was predicted correctly (in a boolean fashion). Obviously, the higher this percentage was, the better the combination of the chosen input parameters. The different test cases for each of the three scenarios are presented and compared to each other in the sequel.

4.4.1 Scenario 1 - No Normalization

4.4.1.1 Comparison of the Labelling Versions

Having analyzed the three versions, their comparison was required so as the best one according to their results to be used. As mentioned in section 4.3, the VOTE version uses only the most frequently appearing label when calculating the centres of the clusters. In this case, it is possible that a label disappears in the created SOM even if it has been used as label within a data sample. For example, if the context that is described by the data samples which have the label 24 Mbps is mapped on cells on which context described by data samples with different labels (36, 48 and 54) has also been mapped with more instances per label than the instances of label 24, then the most frequent label will dominate in this cell. In the case that, this happens for all data samples with the label '24', then this label will not appear on the map. Generalizing this example (with no loss of information), labels with fewer instances will not appear in the created SOM. This causes the elimination of one or more labels and thus the centre of this (these) label(s) won't be calculated. Consequently, cluster(s) that correspond to this (these) label(s) will cease to exist. Moreover, the algorithm will terminate a little after the calculation of the centres as according to the method, the under question data sample is expected to select among four clusters with respect to its distance from the four centres, not less. Finally, even if the algorithm didn't stop, the data, which would be used for the evaluation of the method and would belong to the eliminated cluster(s), would have been correlated with a wrong cluster and label and thus the prediction of the bitrate would be definitely wrong.

Trying to find a solution to the existing problem of VOTE version, ADD1 version was created. In ADD1 version, all possible labels of each cell participate equally and independently of their instances. Both versions were executed using the same data files and the same training parameters. Numerous tests that compare the two versions were performed but for brevity reasons only two indicative examples of them are presented hereafter. Their results are also depicted in Table 4.3. The first example targets at demonstrating the inefficiency of the VOTE version to give results in some test cases and involves the execution of both versions using the data file corresponding to the 3rd row in Table 4.4. When using the VOTE version, the centre of the cluster with label/bitrate equal to 24 was (NaN, NaN), which meant that this cluster ceased to exist and the process terminated reporting an error as there was no centre calculated for the cluster with bitrate equal to 24. On the other hand, when using the ADD1 version, there was one centre for each cluster (24, 36, 48 and 54) and the process resulted with no matlab error in a percentage of correct predictions equal to 52.2%.

Table 4.3: Comparison of the labelling versions for scenario 1

Data File	Percentage of correct predictions		
	VOTE version	**ADD1 version**	**FREQ version**
1	58.100%	56.800%	64.300%
3	-	52.200%	71.400%

The scope of the second example is to present and compare the two versions when both of them give results. Thus, continuing with the second example both versions were also executed using the 1st data file (see Table 4.4). In this test case both versions ended successfully giving results equal to 58.1% and 56.8% of correct predictions for VOTE and ADD1 versions, respectively. Summarizing the above test cases, their result led to the conclusion that ADD1 version solved the problem of VOTE version but, in cases where the latter worked properly, ADD1 version had lower percentage of correct predictions. As a result, when both versions manage to provide results, VOTE version performs better than ADD1 version (see also Table 4.3). However, in both cases, the correctness of predictions did not appear to be very promising.

The above conclusion led us in the creation of FREQ version, which is kind of a hybrid version of the first two. In particular, FREQ version uses all labels of the cells (like ADD1 version does) but contrarily to ADD1 version, labels participate in the calculation of the centres of the clusters unequally, as a weighted average of their frequency. As a result, FREQ version is expected to being able of creating all centres of the four pre-defined clusters, thus always giving results, but also of treating the labels with respect to their appearances/ frequency and not equally. Having created this version, all that was left to be done was its comparison with the first two. In order to present two indicative examples, the 3rd and the 1st data files of Table 4.4 were used for the execution of this version as well. The result of FREQ version, equal to 71.4% when using the 3rd data file and 64.3% when using the 1st data file, was better (with higher percentage of correct predictions of the bitrate) than both the VOTE and the ADD1 versions (Table 4.3). Comparing all the results of Table 4.3, which are only indicative examples of the numerous tests that have been performed, FREQ version proved to give better results in all cases and thus it was selected to be used in the rest of the test cases of this scenario of normalization.

4.4.1.2 Selection of the Variables of Data Samples

The next step in this scenario concerned the variables of the data samples that suit better for predicting the bitrate. In order to do so, many different cases were created. The used version for labelling and calculating the centres of the clusters was FREQ in all test cases. Moreover, the training parameters were kept same as in the tests of sub-section 4.4.1.1. These cases used different data files for both training and evaluation phases. The difference between them lied in the number and the type of the variables of the data samples.

At the created cases there were 8 variables of a data sample that were used in different combinations, namely: RSSI, number of input and output packets, number of input and output errors, number of input and output bytes and bitrate. Table 4.4 depicts the combinations of the variables for each test case and the corresponding data file that contained them. Furthermore, the results of these test cases are also presented in the 9th column (scenario 1) of Table 4.4. The case with the highest percentage of correct predictions, equal to 71.4%, was the one whose variables were the number of input and output packets and RSSI i.e., the test case during which the 3rd data file in Table 4.4 was used. As a result, these variables were also used in the rest of this scenario of normalization.

Table 4.4: Data files and their containing variables (The percentage of correct predictions refers the results obtained when using FREQ version)

Data file	RSSI	Input Packets	Output Packets	Input Errors	Output Errors	Input Bytes	Output Bytes	Percentage of correct predictions		
								Scenario 1	Scenario 2	Scenario 3
1	✓	✓		✓		✓		64.300%	74.100%	71.900%
2	✓	✓	✓	✓		✓		61.900%	75.400%	74.600%
3	✓	✓	✓					71.400%	49.200%	74.600%
4	✓	✓	✓			✓	✓	45.900%	74.900%	74.300%
5	✓					✓	✓	68.100%	73.000%	71.400%
6	✓	✓	✓	✓	✓			32.700%	51.400%	40.000%
7		✓	✓			✓	✓	45.900%	48.400%	56.500%
8	✓	✓						56.500%	43.800%	46.800%

4.4.1.3 Selection of the Number of Data Samples

Having selected the variables of a data sample, the next step includes the decision upon the number of data samples to participate in the training process of SOM. In order to do so, a number of data files, and thus test cases, were created as well. These data files included the variables in which the analysis of sub-section 4.4.1.2 had resulted in (number of input and output packets and RSSI) but with different number of data samples (rows). Once again, the training parameters among the test cases were the same and the FREQ version was used for obtaining results. According to the results (see Figure 4.3), the number of data samples affected the percentage of the correct predictions but not always in the same direction. As can be observed by Figure 4.3, the highest percentage of correct predictions was 73.6% and appeared when the number of data samples was 617. Motivated by this conclusion, the corresponding data file was also used during the remaining test cases of this scenario of normalization.

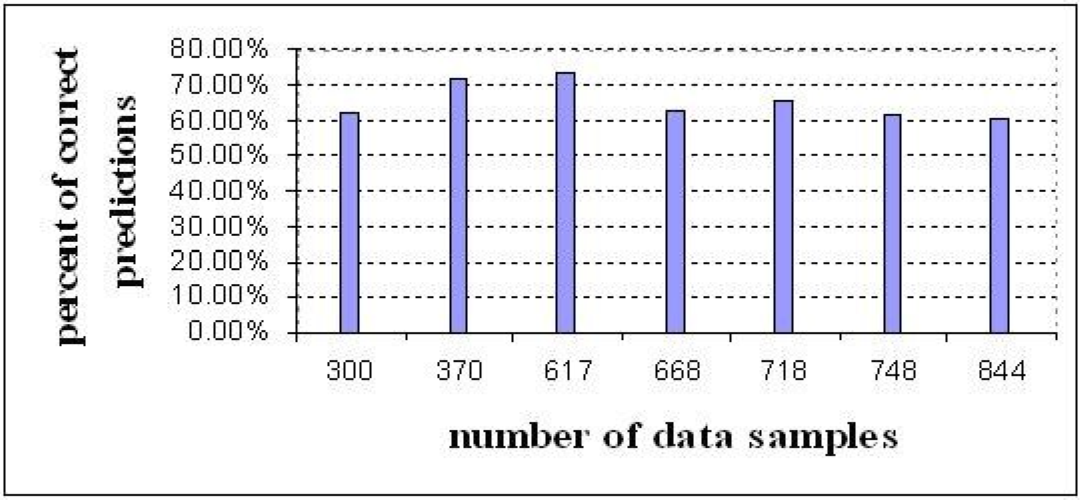


Figure 4.3: Percentage of correct predictions of the bitrate according to the number of the used data samples – Scenario 1

4.4.1.4 Selection of the Parameters per Training Algorithm

The last step during this scenario was, first the decision upon the most appropriate values of training parameters and, then, the selection/ comparison of the two SOM training algorithms.

Towards this direction, different test cases, which for brevity reasons are not analyzed in this section, were tried for each training algorithm. Each test case differed from the previous one only to the value of one parameter (randomly selected). Comparing the results that were derived when using the batch training algorithm, it was obvious that the best set of values of the training parameters was the one shown in Table 4.5. Moreover, Figure 4.4 depicts the predicted values of the bitrate, the real measured values of the bitrate and a comparison among the two above when the batch training algorithm is used and when there is no normalization of the data.

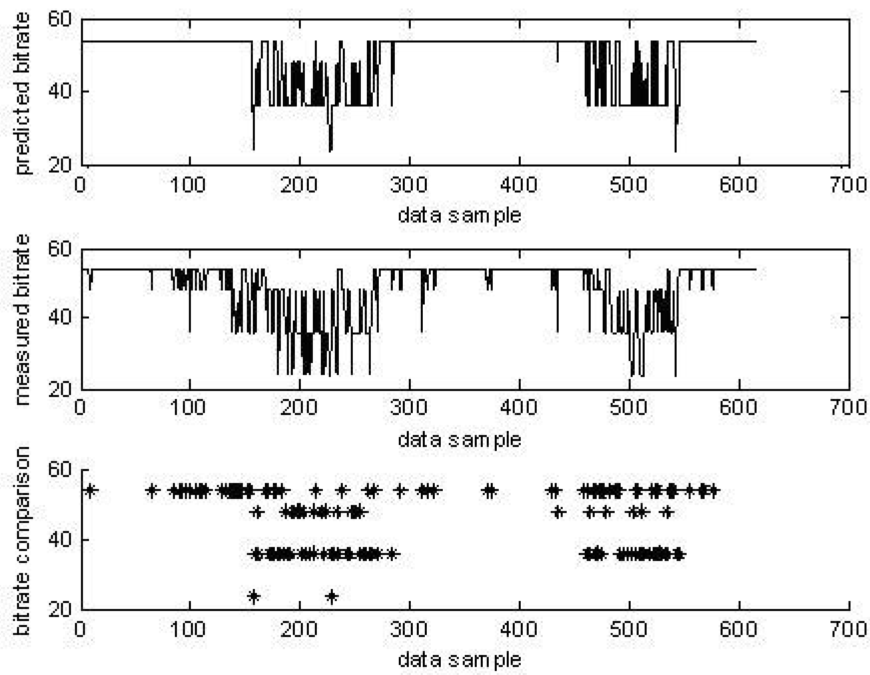


Figure 4.4: Batch Training Algorithm: Predicted bitrate, Measured bitrate and a Comparison of the above. The symbol * depicts only the data samples which have different predicted and real values – Scenario 1

Respectively, in the case of sequential training algorithm, the winning set of values of the training parameters is shown in Table 4.6 and Figure 4.5 depicts the predicted bitrate, the measured bitrate and a comparison of the above in case of sequential training algorithm when there is no normalization of the used data. Although the technique was the same, it's worth mentioning here an important difference: in the sequential training algorithm the samples do not enter the training phase at the same time. As a result, the order with which they are used for training the map may lead to slightly different results. In particular, an already trained cell may be "retrained" as a neighbouring cell of the BMU of another data sample. If these training processes occur with the opposite sequence, the final map will be slightly different. However, this does not provoke SOM from converging but may also lead to different predictions in some cases. In order to avoid such a situation, the entrance of the samples was selected to be always the same and ordered according to the data file.

Table 4.5: Values of the parameters for the batch training algorithm – Scenario 1

Neighbourhood function: Gaussian			
Rough Phase		***Fine-tuning Phase***	
Initial radius	5	**Initial radius**	1
Final radius	1	**Final radius**	1
Training length	6	**Training length**	48

Table 4.6: Values of the parameters for the sequential training algorithm-Scenario 1

Neighbourhood function: Gaussian **Length type:** epochs **Learning function:** inv			
Rough phase		***Fine-tuning phase***	
Initial radius	3	**Initial radius**	1
Final radius	1	**Final radius**	1
Training length	4	**Training length**	21
Initial alpha	0.5	**Initial alpha**	0.05

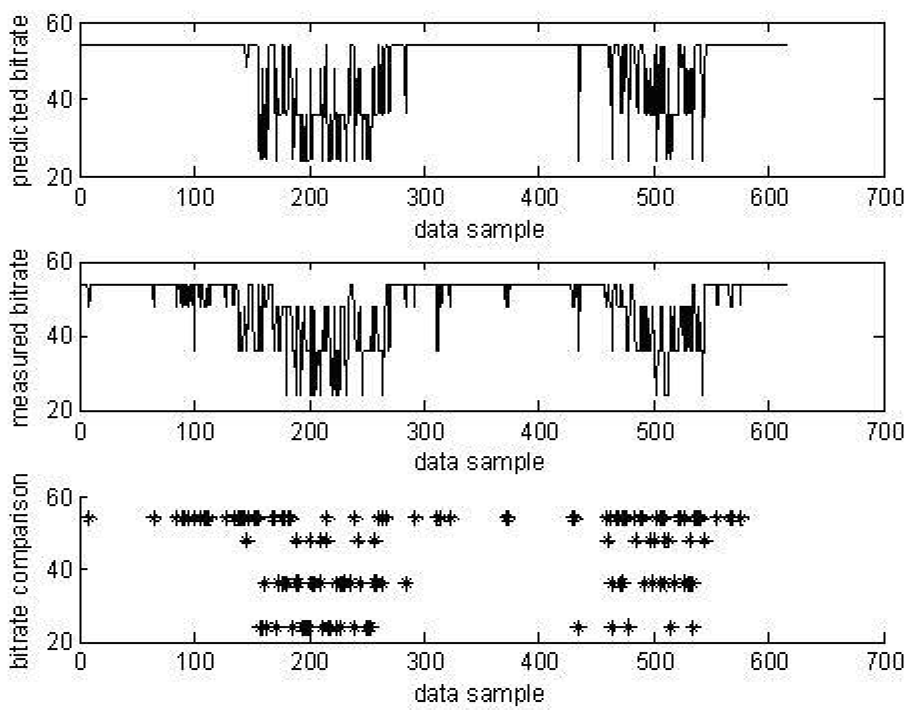


Figure 4.5: Sequential Training Algorithm: Predicted bitrate, Measured bitrate and a Comparison of the above. The symbol * depicts only the data samples which have different predicted and real values – Scenario 1

The comparison of the best result of the batch training algorithm with the one of the sequential training algorithm reveals that the first result, equal to 74.4%, is a little lower than the second one, equal to 75.4%, giving the impression that the best choice is the sequential training algorithm. In addition, according to [1], the memory consumption when using the batch training algorithm can be calculated by the equation

$$MemCon = 8*(5md + 4nd + 2m^2) \qquad (4.6)$$

where m is the number of map units, n is the number of data samples and d is the input space dimension (number of variables in this case). According to the same report, [1], sequential training algorithm requires only one half or one third of the memory capacity needed during batch training algorithm. The corresponding used map size was 130, and the data file comprised of 617 data samples each one of which included 3 variables. Therefore, when the batch training algorithm is used 345232 bytes of memory capacity are needed, whereas about 115077 to 172616 bytes memory capacity is required when the sequential training algorithm is used. On the other hand, the time that is needed to complete the training phase of the SOM is sometimes crucial so it was measured as well. According to the measurements, batch training algorithm is quicker, requiring about 3 to 4 seconds to complete the training phase, while the sequential one requires about the double time i.e. 7-8 seconds. Table 4.7 summarizes the above presented results. As can also be observed through Table 4.7, and because of the fact that the difference between the two results is rather small, the choice between the two algorithms during this scenario of normalization is subjective and depends on the available memory capacity and the existence of the requirement of a quick training or not.

Table 4.7: Comparison of batch and sequential training algorithms in case of scenario 1

	Percentage of correct predictions	**Memory consumption (Bytes)**	**Training Duration (sec)**
Batch Training Algorithm	74.4%	345232	3 – 4
Sequential Training Algorithm	75.4%	115077 – 172616	7 – 8

4.4.2 Scenario 2 – Normalization of the Parameters of Data Samples

4.4.2.1 Comparison of the Labelling Versions

As in the first scenario, where there was no normalization of data, the first tests referred to the selection among the optimal labelling version. For this purpose, all versions were executed using the same data files and the best one was selected with respect to their results. Once again, the selection was based on numerous tests but the results are not depicted here for brevity reasons. In this scenario as well, the best result was taken when using the FREQ version and thus FREQ version was the one that was used for the rest of the test cases of this scenario too.

4.4.2.2 Selection of the Variables of Data Samples

Test cases with different number and type of variables of data samples, using once more the 8 variables of data samples which were used in the first scenario of normalization (RSSI, number of input/output packets, number of input/output errors, number of input/output bytes and bitrate), were also executed. These test cases were the same with those of the first scenario and were using the data files which are depicted in Table 4.4. As can be seen in Table 4.4, during this scenario, the case with the highest percentage of correct predictions, equal to 75.4%, was the one whose variables were the number of input and output packets, the number of input errors, the number of input bytes and RSSI (test case during which the 2nd data file was used). Those variables are the ones that were used for the rest of this scenario.

4.4.2.3 Selection of the Number of Data Samples

Having selected the variables of a data sample, tests with varying number of data samples in the training data file ,but with the same variables, i.e. the variables in which the analysis of sub-section 4.4.2.2 had resulted in (i.e., # of input and output packets, # of input errors, # of input bytes and RSSI), were performed. According to the results, the number of data samples in this scenario as well affected the results of the predictions but not always towards the same direction. The diagram showing these results is depicted in Figure 4.6 while its analysis reveals that the maximum result is

76.5% and appears when the number of data samples is 434. This was also the number of data samples that was used for the next test cases of this scenario.

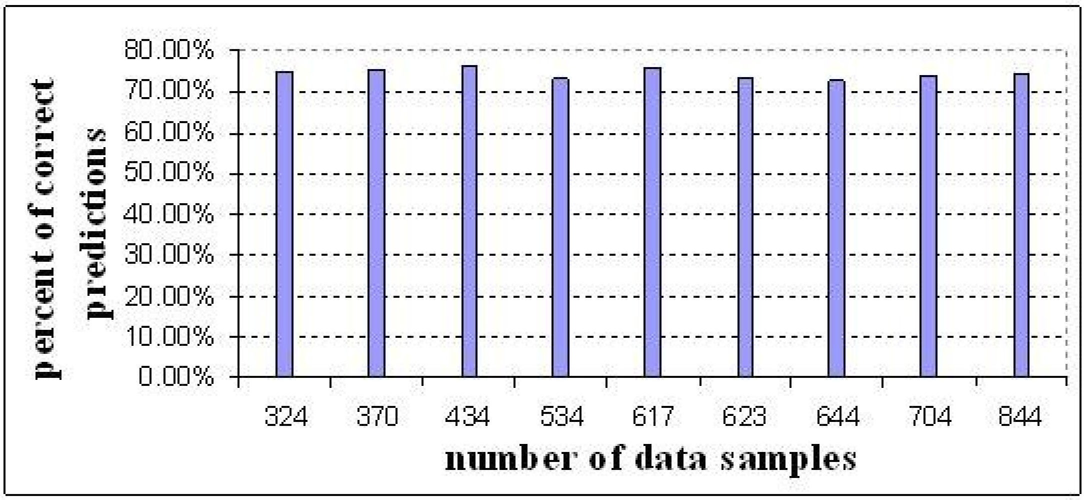


Figure 4.6: Percentage of correct predictions of the bitrate according to the number of the used data samples – Scenario 2

4.4.2.4 Selection of the Parameters per Training Algorithm

Finally, numerous tests took place during the selection of the most suitable training parameters (for each training algorithm) and the comparison between the training algorithms. The procedure that was applied was the same with the first scenario. Comparing the results of batch training algorithm it was obvious that the best choice was the one shown in Table 4.8 while Figure 4.7 depicts a) the predicted values of the bitrate, b) the real measured values of the bitrate and c) a comparison among the two above. Similarly, the best set of training parameter values for the sequential training algorithm is shown on Table 4.9 while Figure 4.8 depicts a diagram with the bitrate predictions, the bitrate real values and their comparison.

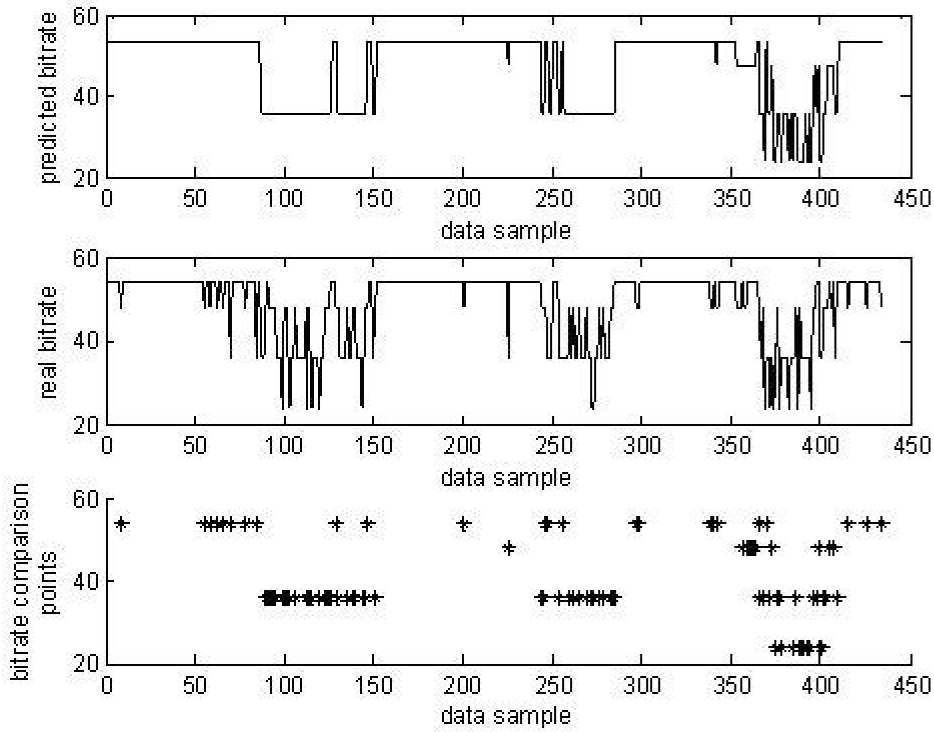


Figure 4.7: Batch Training Algorithm: Predicted bitrate, Measured bitrate and a Comparison of the above. The symbol * depicts only the data samples which have different predicted and real values – Scenario 2

Table 4.8: Values of the parameters for the batch training algorithm-Scenario 2

Neighbourhood function: Gaussian			
Rough Phase		***Fine-tuning Phase***	
Initial radius	3	**Initial radius**	1
Final radius	1	**Final radius**	1
Training length	1	**Training length**	9

Table 4.9: Values of the parameters for the sequential training algorithm-Scenario 2

Neighbourhood function: Gaussian **Length type:** epochs **Learning function:** inv			
Rough phase		***Fine-tuning phase***	
Initial radius	4	**Initial radius**	1
Final radius	1	**Final radius**	1
Training length	5	**Training length**	21
Initial alpha	0.5	**Initial alpha**	0.055

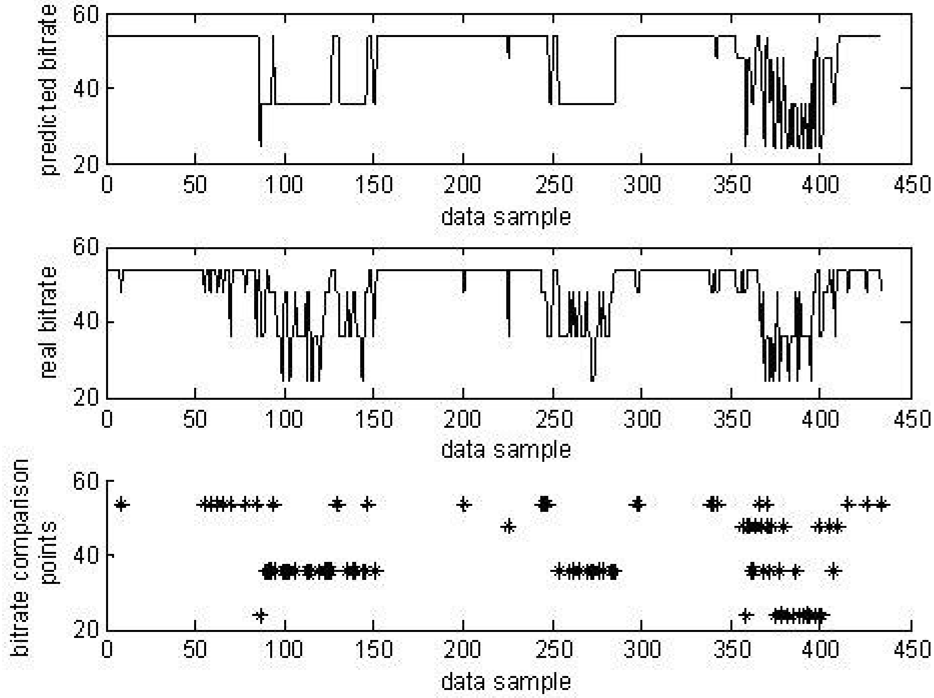


Figure 4.8: Sequential Training Algorithm: Predicted bitrate, Measured bitrate and a Comparison of the above. The symbol * depicts only the data samples which have different predicted and real values – Scenario 2

Accordingly to the first scenario of normalization, the two training algorithms were compared to three significant points: a) to their results, b) to the required memory and c) to the needed training time. The respective values are summarized in Table 4.10.

Table 4.10: Comparison of batch and sequential training algorithms in case of scenario 2

	Percentage of correct predictions	**Memory consumption (Bytes)**	**Training Duration (sec)**
Batch Training Algorithm	77.4%	266840	< 1
Sequential Training Algorithm	77.4%	88946 – 133420	4 – 5

Taking into consideration all three observations, the best choice among the training algorithms in this scenario is certainly the batch training algorithm in cases where a quick training is needed and a good memory capacity is available. Alternatively, i.e. if the time is not crucial and/ or the available memory capacity is not enough, sequential training algorithm can provide equally good results, as well.

4.4.3 Scenario 3 – Normalization of the Variance of Data Samples

4.4.3.1 Comparison of the Labeling Versions

As well as in the first two scenarios of normalization, after the execution of all versions of the method with the same data files, the best one was selected with respect to the percentage of correct predictions of bitrate, for each version. Although the results of the numerous tests are not explicitly mentioned here, once more, the highest percentage of correct predictions was received by the FREQ version and thus this was the one that was used during this scenario as well.

4.4.3.2 Selection of the Variables of Data Samples

This phase of the scenario included again test cases during which the SOM was trained using the data files of Table 4.4. During this scenario, the highest percentage of correct predictions, as can be observed by Table 4.4, was equal to 74.6% and appeared in two test cases. The variables of the data samples of the first test case were the number of input and output packets and the RSSI (data file 3) while the second one included the number of input errors and the number of input bytes as well (data file 2). However, the test cases were further compared to each other with respect to the memory that would be required in case of applying data file 2 and 3, respectively. The used values with respect to the map size, the number of data samples and the number of the variables can be found in Table 4.11. Moreover, in the same table (last column) can be found the memory consumption that derives from this values when applying equation (18). Among the two test cases, the second one demands higher memory capacity than the first one.

Table 4.11: Comparison of test cases 2 and 3

Data File	Map Size (m)	Number of Data Samples (n)	Number of Variables (d)	Memory Consumption (Bytes)
2	98	370	3	200944
3	98	372	5	232464

As a result, the optimal choice between these two was the first one. Its variables (number of input and output packets and RSSI) were the ones that were used in the rest of this scenario.

4.4.3.3 Selection of the Number of Data Samples

In the sequence, tests with different number of data samples during the training process were executed. These tests were only varying to the size of the data file and definitely not to the variables that comprised the data samples. On the contrary, the used variables were the ones in which the analysis of sub-section 4.4.3.2 had resulted in, i.e., the number of input and output packets and the RSSI. According to the results, each increase of the number of data samples affected the results in a different way. The percentage of correct predictions of bitrate for each test case is depicted in the diagram of Figure 4.9 while the analysis of the latter reveals that the highest percentage was 76.9% and appeared when the number of data samples was 668. This is also the number of data samples that were used for the next test cases of this scenario.

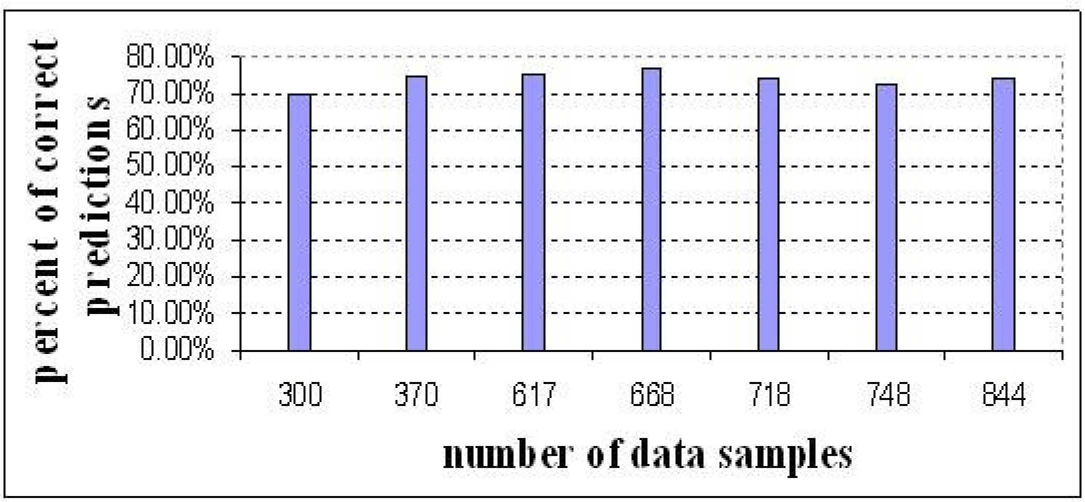


Figure 4.9: Percentage of correct predictions of the bitrate according to the number of the used data samples-Scenario 3

4.4.3.4 Selection of the Parameters per Training Algorithm

Finally, numerous tests were performed so as the best value of each parameter of both of the training algorithms to be defined. As soon as the optimal values of the training parameters of each training algorithm were selected, the comparison of the latter took place. The followed process is already known from the previous normalization scenarios: only one random parameter of Table 4.1 or Table 4.2 was changing at a time. When changing the value of this parameter resulted in no or slight alterations of the result, another random parameter was selected until all parameters had been examined. The best set of values of the parameters in the case of batch and sequential training algorithms are shown in Table 4.12 and Table 4.13, respectively, while Figure 4.10 and Figure 4.11 depict accordingly a) the predicted values of the bitrate, b) the real measured values of the bitrate and c) a comparison among the two above under this set of values.

Table 4.12: Values of the parameters for the batch training algorithm-Scenario 3

Neighbourhood function: Gaussian			
Rough Phase		***Fine-tuning Phase***	
Initial radius	4	**Initial radius**	1
Final radius	1	**Final radius**	1
Training length	2	**Training length**	10

Table 4.13: Values of the parameters for the sequential training algorithm-Scenario 3

Neighbourhood function: Ep **Length type:** epochs **Learning function:** inv			
Rough phase		***Fine-tuning phase***	
Initial radius	3	**Initial radius**	1
Final radius	1	**Final radius**	1
Training length	8	**Training length**	20
Initial alpha	0.5	**Initial alpha**	0.05

Finally, the comparison of the training algorithms in this scenario of normalization in the context of their results, the required memory and the needed training time resulted in Table 4.14. As a result, when the variance of data samples is normalized, batch training algorithm is clearly the optimal choice if enough memory capacity is available. Alternatively, sequential training algorithm should be used.

Table 4.14: Comparison of batch and sequential training algorithms in case of scenario 3

	Percentage of correct predictions	**Memory consumption (Bytes)**	**Training Duration (sec)**
Batch Training Algorithm	78.9%	333264	< 1
Sequential Training Algorithm	77.5%	111088 – 166632	7 – 8

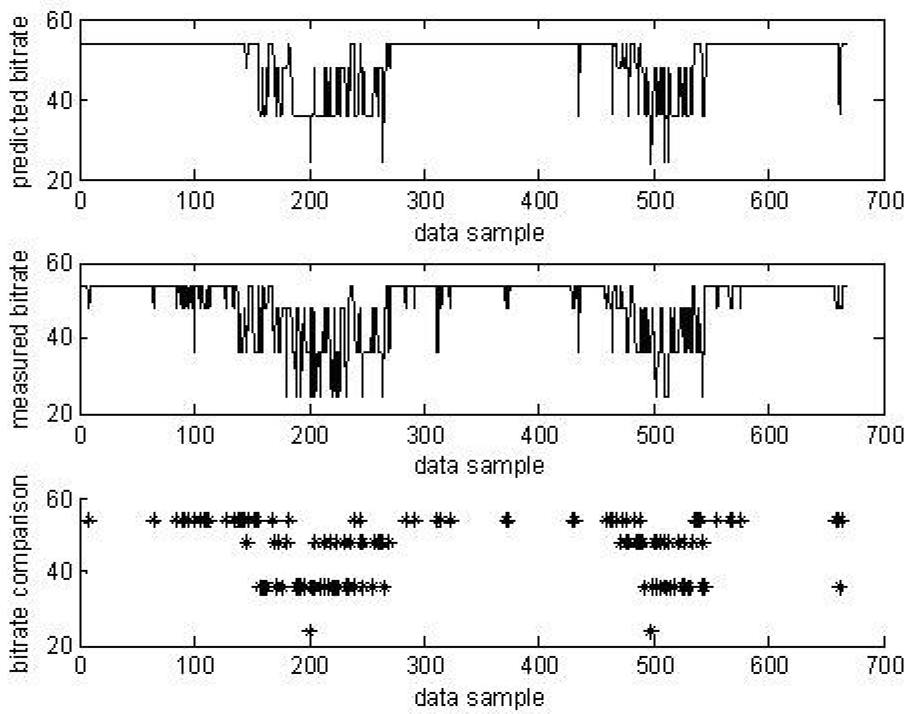


Figure 4.10: Batch Training Algorithm: Predicted bitrate, Measured bitrate and a Comparison of the above. The symbol * depicts only the data samples which have different predicted and real values-Scenario 3

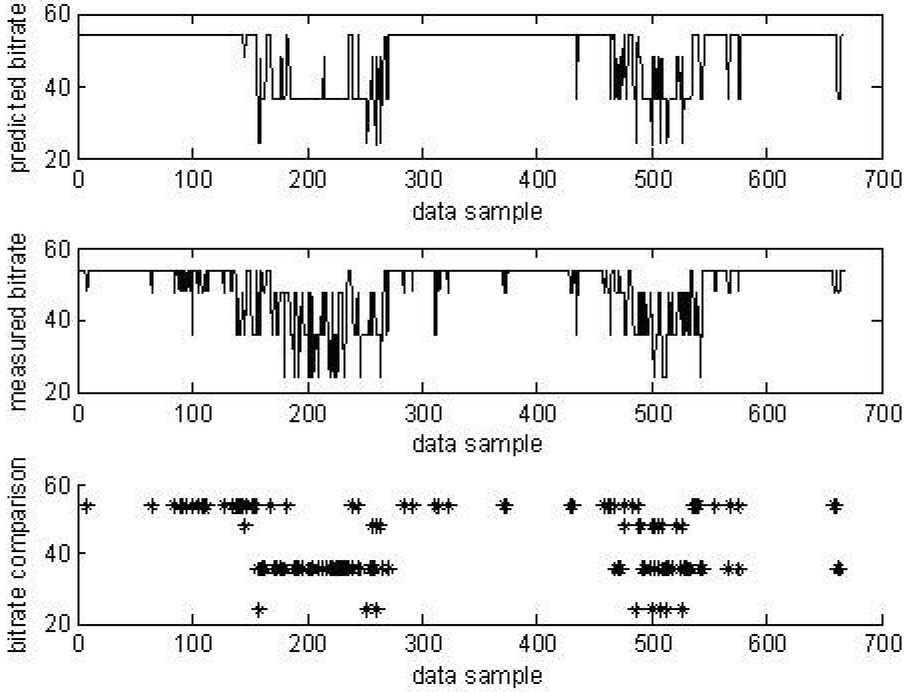


Figure 4.11: Sequential Training Algorithm: Predicted bitrate, Measured bitrate and a Comparison of the above. The symbol * depicts only the data samples which have different predicted and real values-Scenario 3

4.4.4 Comparison of the Scenarios of Normalization

Comparing the three scenarios, the last two scenarios demonstrate the highest percentage of correct predictions equal to 77.4 and 78.9 for the second and the third scenario, respectively. Moreover, since these percentages are quite close, further comparing points were needed before concluding to the optimal normalization scenario/ method. The additional comparing points were selected to be a) the number of data samples used for the training, b) the size of the map, c) the number and the type of the variables of the data samples and d) the memory consumptions. The comparison of the scenarios in the above mentioned points is summarized in Table 4.15 and reveals a number of data samples versus number and type of variables trade off. Thus, and because of the fact that the difference of the two percentages is rather small, the choice between these two scenarios is subjective. On the other hand, in cases of limited memory capacity, scenario 2 is preferred as the required memory is 266.8 Mbytes versus 333.2 Mbytes of the third scenario.

Table 4.15: Comparison of the scenarios of normalization

Scenario	**Percentage of correct predictions**	**Number of Data Samples**	**Map Size**	**Number of variables**	**Memory Consumption (Mbytes)**
1a	74.4%	617	130	3	345.2
1b	75.4%	617	130	3	115.1-172.6
2	77.4%	434	105	5	266.8
3	78.9%	668	126	3	333.2

4.5 Comparison with Supervised NN-Based Approach

In this sub-section, specific supervised, NN-based techniques that were presented and assessed in [2] and [4] were chosen, so as to act as benchmarks for the above proposed SOM-based method.

In particular, the solution proposed in [2] argues to assist the cognitive radios in selecting the desired radio configuration by predicting the achieved data rate of a set of candidate radio configurations. The study presents and evaluates two learning schemes, which are based on NNs namely, the basic and the extended one. The basic one referred to an Elman network [3] which was a two-layer back-propagation, recurrent network, with the addition of a feedback connection from the output of the unique hidden layer to the input layer. During the evaluation, the best test case was proved to be based on normalized data in the range of [0, 1] and on a training session that lasted for 300 epochs and used a set of 3000 training data. Finally, the Mean Square Error (MSE), which was the performance evaluation metric of this test case was equal to 0.0100 when the training and the validation data set were the same. The extended NN-based learning scheme comprised test cases using FF back-propagation, Elman, FTDNN and custom FF back-propagation NNs while the data of this scenario were normalized in the range of [-1, 1]. The lowest MSE for this scenario, equal to 0.0549 when the validation data set was a subset of the training data set, arose from the test case whose training lasted for 300 epochs and used a set of 2400 training data. Finally, it's worth mentioning that all the used learning schemes were supervised.

The work in [4] comprises also results from research that focuses on predicting the data rate according to the past knowledge of a network using NN-based schemes. In this research three scenarios are analyzed. In the first scenario the data rate is predicted based on past measurements. These measurements were firstly normalized in the range of [-1, 1]. Moreover, there are two types of NNs that were examined: the FTDNN and the Elman NN. Once again, the MSE was used as the main performance evaluation metric for the scenario. The best result taken from this scenario, when the evaluation data set was a subset of the training data set, was equal to 0.0194 and evoked in a test case of FTDNN whose training lasted for 100 epochs and the set of training data was 50000. During the second scenario, the prediction of the data rate is based on enhancing the input layer with variables like the quality of the link and the signal strength of the wireless transceiver. All test cases that had been conducted used normalized data in the range of [-1, 1] and FTDNN. In this case, the best available network design pattern was the one whose training session had lasted for 10 epochs and had used a set of 5000 training data. The MSE, produced when the validation data was a subset of the training data, was 0.0468. Finally, the last scenario concerned the prediction of the actual achieved throughput in a short term fashion in environments that were rapidly changing. In order to do so, two sub-scenarios were distinguished. In both sub-scenarios, the prediction was based on the RSSI and the number of bytes transmitted but the prediction of the first sub-scenario concerned the instantaneous expected throughput while the prediction of the second sub-scenario concerned the average expected throughput, in a specific short period of time. Moreover, in both sub-scenarios, different types of NNs (including Elman, linear and FF networks) were tested and the best available network design pattern was a FTDNN whose training session had lasted for 20 epochs and had used a set of 700 training data. Finally, in case of sub-scenario 1, where the validation data set was a subset of the training data set, the MSE was equal to 0.0156 while in case of sub-scenario 2, where the validation data set was a subset of the training data set, the MSE was equal to 0.0092.

As also mentioned in sub-section 4.4, the metric for the results of this study was the number of data samples whose bitrate was predicted correctly (expressed in percentage units). This metric is rather quantitative comparing to MSE. In order to have a qualitative metric as well and to be able to compare the results of the proposed method with the results of the above mentioned researches the MSE of the best scenarios of this study (scenario 2 - normalization of the parameters of data samples and scenario 3 - normalization of the variance of data samples) were also calculated. The MSE of the best case of scenario of normalization of the parameters of data samples is equal to 0.0346 while the MSE of the best case of scenario of normalization of the variance of data samples is equal to 0.0312.

Table 4.16: Comparison with other NN-based techniques (supervised)

Scenario / Paper	Training Data Set	Epochs	MSE
1st scenario of [2]	3000	300	0.0100
2nd scenqario of [2]	2400	300	0.0549
1st scenario of [4]	50000	100	0.0194
2nd scenario of [4]	5000	10	0.0468
1st sub-scenario of 3rd scenario of [4]	700	20	0.0156
2nd sub-scenario of 3rd scenario of [4]	700	20	0.0092
2nd scenario of the dissertation (section 4.4.2)	434	1+9=10	0.0346
3rd scenario of the dissertation (section 4.4.3)	668	2+10=12	0.0312

Table 4.16 summarizes the above mentioned information regarding the number of the data samples, the training epochs and the MSE for each scenario. This information is also exploited for comparing the considered approaches. To begin with, the basic comparing point was the achieved MSE per approach. Comparing the results of this approach with those of [2] and [4], there are 4 scenarios whose MSE is lower than that of the studied scenarios in sub-section 4.4. These scenarios are the 1st one of [2], the 1st one of [4] and the two sub-scenarios of the 3rd scenario of [4].

On the other hand, in all cases (apart from the 4th one, i.e. 2nd sub-scenario of the 3rd scenario in [4], which in any case has higher MSE than that of the proposed method) the epochs needed for training the corresponding NN, i.e., for building the knowledge, were significantly more than the epochs required for training the SOM. This significant difference of required epochs (time) for the training proves to be of high importance when a method is destined for online training. It should be noted here that when calculating the number of the epochs in the case of the proposed method, both rough and fine-tuning phases should be taken into account. For example, in the case of the 2nd scenario of this study, the number of epochs is 1 for the rough phase and 9 for the fine-tuning phase resulting in 10 epochs in total.

Moreover, when comparing the scenarios with respect to the number of the data samples, it appears that in all NN-based schemes more training data are required for achieving so low MSE. This turns out to be a disadvantage of these schemes in the lack of training data. Finally, the proposed approach enforces the possibility of using more variables without demanding changes on the network design pattern.

Summarizing the above conclusions, even if there are cases where SOM demonstrates worst results, in terms of higher deviation (MSE) than other supervised NN-based schemes, the former performs better in terms of needing less training data and epochs and also offering higher design flexibility.

4.6 Comparison with Bayesian statistics Approach

This sub-section compares the SOM-based approach studied in sub-sections 4.3-4.4 withe the Bayesian statistics approach proposed in [8]. In this approach, the mechanism is based on the correlation of candidate transmitter configurations with the QoS, in terms of bit rate, that is offered by the network given this configuration. In particular, the learning mechanism exploits the knowledge and the past experience by enforcing them with Bayesian statistics techniques suitable for reasoning about probabilistic relationships [9]-[11].

More specifically, since the goal is to associate different configurations of a transmitter with the bitrate, the probability to obtain a specific network capacity BR_i given the configuration CFG_i is calculated. This calculation and its frequent update constitute the basis of this technique. The update of these relies on approaches suggested in [9] and [11]-[13].

To begin with, using the Shannon theorem and gathering the necessary information for each configuration makes it possible to calculate the available bit rate for each configuration. Furthermore, using all possible combinations of configurations and bit rate, conditional probabilities of the form $Pr_{kj}[BR_k|CFG_j]$ can be calculated. These probabilities are then organised in conditional probability tables (CPTs) of the form of Table 4.17. In these CPTs each column represents a different configuration while each row corresponds to a different reference value of bit rate. These reference bit rates comprise the set of available BRs which in this case was selected to be discrete [14].

Table 4.17: Example of CPT

BR	**....**	***cfg$_i$***	**....**
br_1	....	$\Pr[BR = br_1 \mid CFG = cfg_i]$	....
br_2	....	$\Pr[BR = br_2 \mid CFG = cfg_i]$	....
....	....	....	....
br_j	....	$\Pr[BR = br_j \mid CFG = cfg_i]$	....
....	....	....	....
$br_{\|M\|}$	....	$\Pr[BR = br_{\|M\|} \mid CFG = cfg_i]$	....

Finally, using the CPTs, the paper identified the most probable bit rate given a configuration by associating to the highest conditional probability in the respective configuration column.

Comparing the two approaches presented for learning network capabilities, Bayesian statistics offer also the possibility of online training. The latter abstracts the necessity of explicitly storing past observations, a feature that doesn't exist in SOM. As a result, the approach minimises the required memory capacity and thus it could also be applied from a user device perspective. If so, then the device would become for example capable of certifying that the offered network capability indeed reaches the value that is claimed by the network.

4.7 Conclusions

Going through numerous test cases the achieved correct predictions of the bitrate reached the percentage of 78.9% of the tested data samples. Such a method is expected to assist CRS to choose among the different candidate configurations by taking into account the predictions of the bitrate that can be achieved. Finally, conducted experiments revealed that there exist cases in which the proposed method exhibits better performance when compared to some supervised NN-based learning schemes available in the literature, while at the same time, the use of SOM technique in the proposed method offers the native flexibility in changing the number and/or the type of the variables without demanding changes in the network design pattern. Bayesian statistics can offer a more light-weighted approach in terms of storage requirements and thus complementary from the user device side.

4.8 Chapter References

[1] J. Vesanto, J. Himberg, E. Alhoniemi, J. Parhankangas, "SOM Toolbox for Matlab 5 Technical Report A57", Neural Networks Research Centre, Helsinki University of Technology, Helsinki, Finland, 2000

[2] K. Tsagkaris, A. Katidiotis, P. Demestichas, "Neural Network-based Learning schemes for Cognitive Radio systems", Computer Communications, Vol. 31, Issue 14, pp. 3394-3404, 2008

[3] J. L. Elman, "Finding Structure in Time", Cognitive Science 14: 179-211, 1990

[4] A. Katidiotis, K. Tsagkaris, P. Demestichas, "Performance Evaluation of Artificial Neural Network-Based Learning Schemes for Cognitive Radio Systems", Computers & Electrical Engineering, Volume 36, Issue 3, Pages 518-535, May 2010

[5] A. Bantouna, K. Tsagkaris, P. Demestichas, "Self-Organizing Maps for Improving the Channel Estimation and Predictive Modelling Phase of Cognitive Radio Systems", in Proc. 20th International Conference on Artificial Neural Networks (ICANN 2010), Thessaloniki, Greece, September 15-18, 2010

[6] R. Céréghino, Y. – S. Park, "Review of the Self-Organizing Map (SOM) approach in water resources: Commentary", Elsevier, Environmental Modeling & Software 24 945-947, 2009

[7] Y.-S. Park, J. Tison, S. Lek, J.-L. Giraudel, M. Coste, F. Delmas, "Application of a self-organizing map to select representative species in multivariate analysis: a case study determining diatom distribution patterns across", France, Ecological Informatics 1, 247-257, 2006.

[8] P Demestichas, A Katidiotis, K Tsagkaris, E Adamopoulou, K Demestichas, Enhancing channel estimation in cognitive radio systems by means of Bayesian networks. Wirel. Personal Commun. 49(1), 87–105 (2009). doi:10.1007/s11277-008-9559-1

[9] RE Neapolitan, Learning Bayesian Networks (Prentice-Hall Series in Artificial Intelligence, NJ, 2003)

[10] J Pearl, Probabilistic Reasoning in Intelligent Systems (Morgan Kaufmann, San Francisco, 1988)

[11] F Jensen, TD Nielsen, Bayesian Networks and Decision Graphs (Springer, NY, USA, 2001)

[12] WM Bolstad, Introduction to Bayesian Statistics (John Wiley & Sons, NJ, 2007)

[13] SJ Russell, P Norvig, Artificial Intelligence: A Modern Approach (Prentice-Hall, NJ, 2002)

[14] R Barco, P Lázaro, L Díez, V Wille, Continuous versus discrete model in auto-diagnosis systems for wireless networks. IEEE Trans. Mob. Comput.7(6), 673–681 (2008). doi:10.1109/TMC.2008.23

5 COGNITIVE DATA ANALYTICS OF WIRELESS WIDEBAND INFRASTRUCTURES

5.1 Chapter Outline

This chapter presents mechanisms related to the analysis of data related to the traffic of either a core or an access wireless network. In particular, two mechanisms are proposed and evaluated hereafter. The first mechanism is building knowledge on the traffic of a core network and predicts how possible it is to run on a congested link. The data exploited in this mechanism are directly monitored by the network, i.e., refer to network parameters, and thus allow narrow timeslots for proactively overcoming a congested link. On the other hand, the second mechanism builds knowledge with respect to the traffic of the access network but takes into account human-oriented parameters (e.g., time, date, location, etc.) as well, allowing to also foresee more long-term situations and adjust the network parameters accordingly.

5.2 Prediction of Congestion Levels of a Core Network Link

5.2.1 Problem statement

Congestion is only one of the reasons that can cause failure to transfer data and is highly connected to the number of the users and the types of services that are willing to use. However, the trend of a link towards congestion can also be implied by other things, e.g., increases on the rate of losing packets in a specific link. Other such variables are traffic load of the selected area (in terms of either number of sent packets or Bytes or number of users), queue length and time that is required for a packet to leave the node (delay). Empirically, we can assume that an increase on the values of these variables brings the link closer to congestion but we cannot exactly evaluate the limits of congestion (e.g., how far from or close to congestion the link is or even if it is already congested), especially when variables that are concerned are more than one.

Additionally, variables such as the used window size, the available buffer and of course the capacity of the link also influence these limits in the opposite direction.

The study presented in this sub-section of the dissertation refers to a mechanism that builds knowledge on the relationship between these variables and exploit it for predicting the possibility of congestion under a context described through these variables. In particular, the mechanism is based on the unsupervised learning technique of SOMs (see section 3.4) and its function has two phases: a) pattern recognition of congestion with respect to a combination of the above described variables (some or all of them) and b) prediction of congestion.

In particular, the past events of congested links are used so as to model representative data and define reusable congestion patterns, for micro or macro events. Towards this direction event/parameter correlation algorithms were the key of such a process.

5.2.2 The technique

During the first phase of the mechanism, pattern recognition of congestion, SOM was used for training maps to depict the relationship between the variables which describe the context and the respective possibility of congestion. For making visible the congestion of the data samples on the map, labels and colours, that reveal how close to congestion the link is, were used.

Moving to the next phase where the prediction of the congestion takes place (Figure 5.1), a new data sample, the congestion possibility of which is to be predicted, can be mapped on the recognized pattern of the data using again SOM. When the data sample is mapped, and due to the ability of SOM to map similar data samples close to each other, the new data sample is expected to be similar to those that are mapped to neighbouring cells. Thus, its congestion level is expected to be the same with most of them.

A mathematical model for transforming the visual incentive to content that is able of being recorded or exchanged through a message is required. Towards this direction an approach that involves the categorization of the data sample only according to its first (cells that contact the BMU) or second neighbours (cells that contact the first neighbours) was followed. According to this approach, the data sample belongs to that cluster where most of the surrounding cells belong to.

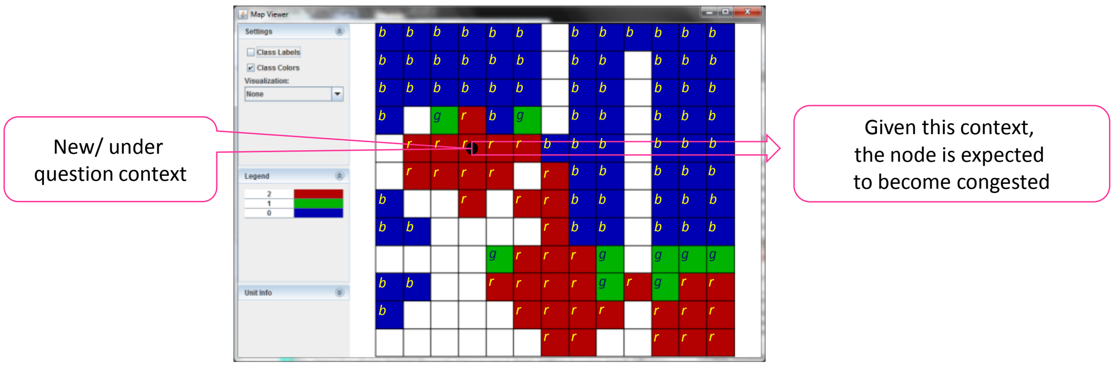


Figure 5.1: Overview of congestion prediction phase of the mechanism.

Finally, for the evaluation of the technique, unseen data, in terms of not have been used during the training of the map and pattern recognition, were used. The congestion level given this context was predicted and the predictions were compared to the real state of the system. The metric that was used for the evaluation of the technique was the percent of correct predictions.

5.2.3 Results

The technique that was developed and applied was based on a hybrid SOM, and particularly on a Parameterless-Growing SOM (PLGSOM). Moreover, the data that were used included some or all of the variables shown in Table 5.1, some of which were directly monitored while others of them need pre-processing for retrieving their values.

However, in the absence of real data, the data, that were used, were derived from simulating a topology with Network Simulator version 2 (NS-2). More specifically, the simulated topology is the one depicted in Figure 5.2. In this topology, the main traffic comes from VoIP services and TCP and UDP packets.

Table 5.1: Variables that were/could be used for the tests

Variable	***Description***	***Type of data***
$BytesIn$	incoming to the link Bytes in the timeslot dt_i	Monitored data
$d(BytesIn)$	trend (derivative) of incoming to the link Bytes, $d(BytesIn)_i = BytesIn_{t_i} - BytesIn_{t_{i-1}}$	Pre-processed data
$dd(BytesIn)$	Second derivative of incoming to the link Bytes, $dd(BytesIn)_i = d(BytesIn)_i - d(BytesIn)_{i-1}$	Pre-processed data
$LinkCap$	link capacity in Mbps	Monitored data
$LinkU$	Required link capacity for serving all traffic, $LinkU_i = \frac{\frac{BytesIn_i \times 8}{dt_i}}{LinkCap_i \times 10^6}$	Pre-processed data
$d(LinkU)$	trend (derivative) of $LinkU$, $d(LinkU)_i = LinkU_i - LinkU_{i-1}$	Pre-processed data
$dd(LinkU)$	Second derivative of $LinkU$, $dd(LinkU)_i = d(LinkU)_i - d(LinkU)_{i-1}$	Pre-processed data
$PktsIn$	incoming to the link packets for the timeslot dt_i	Monitored data
$d(PktsIn)$	trend (derivative) of incoming to the link packets, $d(PktsIn)_i = PktsIn_i - PktsIn_{i-1}$	Pre-processed data
$dd(PktsIn)$	Second derivative of incoming to the link packets, $dd(PktsIn)_i = d(PktsIn)_i - d(PktsIn)_{i-1}$	Pre-processed data
$AvQSize$	Average number of packets (queue size) in the	Monitored data

	buffer during dt_i, $AvQSize_i = \frac{\int_{t_{i-1}}^{t_i} QSize(t)dt}{dt_i}$	
$d(AvQSize)$	trend (derivative) of queue size $AvQSize$, $d(AvQSize)_i = AvQSize_i - AvQSize_{i-1}$	Pre-processed data
$dd(AvQSize)$	Second derivative of $AvQSize$, $dd(AvQSize)_i = d(AvQSize)_i - d(AvQSize)_{i-1}$	Pre-processed data
$MaxQSize$	Maximum queue size before first drop occurs, i.e. buffer size	Monitored data
$QueueU$	Queue Utilization in terms of percent of buffer in use, $QueueU_i = \frac{AvQSize_i}{MaxQSize_i}$	Monitored data
$d(QueueU)$	trend (derivative) of queue utilization, $d(QueueU)_i = QueueU_i - QueueU_{i-1}$	Pre-processed data
$dd(QueueU)$	Second derivative of queue utilization, $dd(QueueU)_i = d(QueueU)_i - d(QueueU)_{i-1}$	Pre-processed data
$Drops$	Number of dropped packets in timeslot dt_i	Monitored data
$DropRatio$	Percent of dropped packets in dt_i, $DropRatio_i = \frac{Drops_i}{PktsIn_i}$	Monitored data
$d(DropRatio)$	trend (derivative) of $DropRatio$, $d(DropRatio)_i = DropRatio_i - DropRatio_{i-1}$	Pre-processed data

$dd(DropRatio)$	Second derivative of $DropRatio$, $dd(DropRatio)_i = d(DropRatio)_i - d(DropRatio)_{i-1}$	Pre-processed data
$conLvl$	Congestion Level of the link, in terms of labeling how congested the link is: $conLvl_i = \begin{cases} 0, & DropRatio_i = 0 \\ 1, & 0 < DropRatio_i < 0.1 \\ 2, & DropRatio_i \geq 0.1 \end{cases}$	Monitored data
$nextConLvl$	Congestion level of the next timeslot dt_i, $nextConLvl_i = conLvl_{i+1}$	Pre-processed data

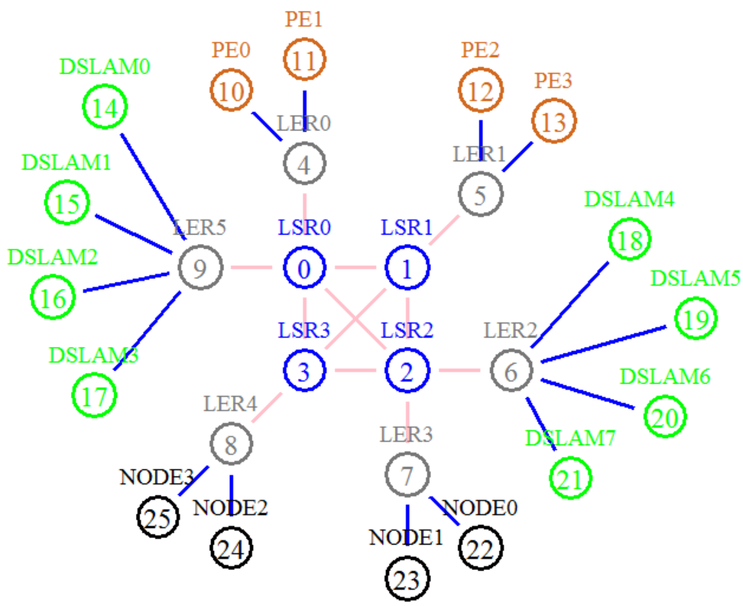


Figure 5.2: Network topology which was used during the simulation.

The under question regarding its congestion link is the link between the nodes 0 and 2, i.e., LSR0 and LSR2 and thus the monitored data refer to data incoming to node 0 (LSR0) with destination to node 2 (LSR2), attributes of node 0 or the link itself.

Using the data that were derived from this simulation and different combinations of the above described variables (Table 5.1) different test cases were created. Among these the best performance was achieved when $BytesIn$, $d(BytesIn)$, $LinkCap$, $AvQSize$, $MaxQSize$ and $DropRatio$ were used. It is also important to note that the congestion level that was used as labelling for the visualization of the data sample on the map was selected to refer to the next timeslot since our objective was to predict the congestion and thus the next state of the system. The respective map that was created when training SOM can be shown in Figure 5.3 while the corresponding percent of correct predictions is 86.6%.

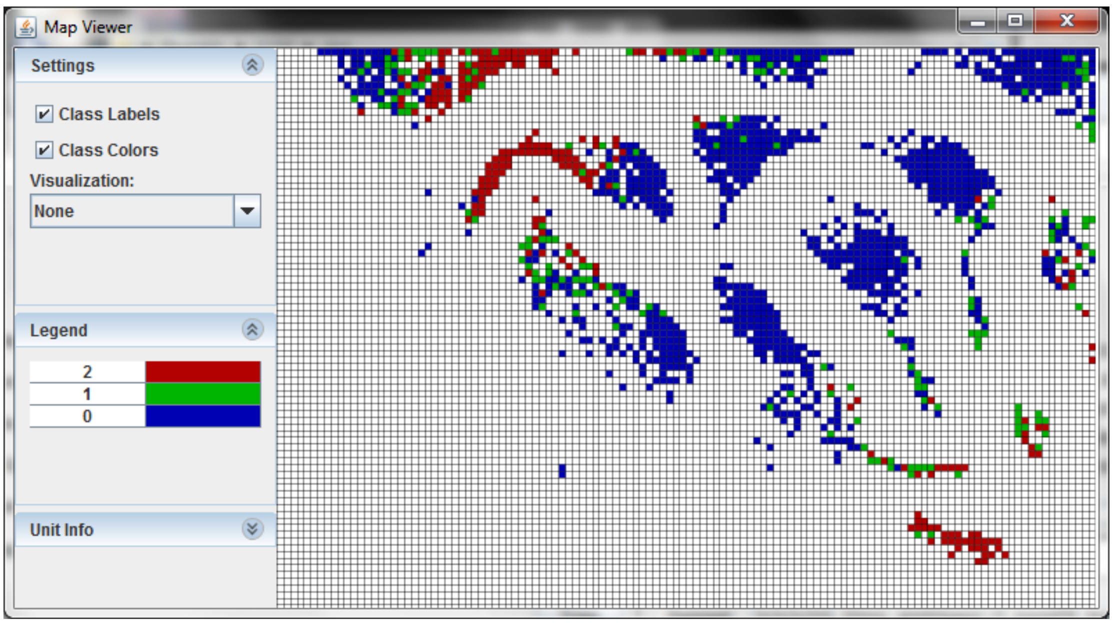


Figure 5.3: SOM depicting the congestion levels (0 in blue labels when the link can serve all the traffic, 1 in green labels when some packets drop but yet is not treated as a congested link and 2 in red labels when the link is expected to become congested) of the under question link

5.3 Prediction of Network Traffic

5.3.1 Problem Statement

This study addresses 2 challenges:

1. the need for a mechanism that will analyze Big Data, build knowledge upon them and decrease the data that will eventually need to be stored;
2. the need for short- and long-term predictions of network load in order to enhance decision making mechanisms that dynamically plan the resources of the network and manage it.

Towards this direction, the study uses the unsupervised machine learning technique of SOMs (see sub-section 3.4) in order to build knowledge on network load and reduce the storage requirements, i.e., the volume of the data that will need to be stored. In particular, according to the SOM theory (see also section 3.4), the SOM technique maps multi-dimensional data onto 2D maps. This means that the dimensions of the data decrease from many to 2.

It also proposes a mechanism that exploits this knowledge to predict network load in the near and distant future. SOMs, as defined by T. Kohonen, receive multi-dimensional data, i.e., data that come from diverse sources and clusters them based on their similarity. This clustering eventually allows the mechanism to keep only some representative data and discard the rest without any loss of information. A mechanism that exploits this knowledge to predict the load that will be encountered in the near or distant future is proposed and validated.

In other words, using a SOM is expected to enhance Big Data management in terms of (a) handling data from different sources, composing and treating them as data samples; (b) decreasing the amount of data during the transition of multiple dimensions to two when representing the multi-dimensional data on 2D maps; and (c) decreasing the storing requirements since a representative sample of the data (only the most informative instances, i.e., instances that are very different to the currently observed ones) along with the clusters of the map are enough to maintain all the information.

In order to also support long term predictions of load, the mechanism builds knowledge using not only network parameters but parameters coming from other sources as well. In particular, network load patterns are expected to be related to the area in which the network load is observed and shows the time, the day, the weather, and the features of the days. For example, in southern countries the load in an entertainment area will probably decrease on a rainy day as most people would prefer staying at home rather than going outside. Moreover, user habits and thus, load patterns may differ among countries, e.g., in northern countries, user preferences for going out (in an entertainment area) may not be directly influenced by the weather. Accordingly, load would be expected to decrease in business areas during a weekend or a (bank) holiday while load in domestic or entertainment areas would most likely increase in similar cases. The above statements/examples are only qualitative estimations and do not necessarily apply in all cases. Therefore, more quantitative study and validation is needed to also facilitate the use of the predictions in dynamic network management and resource planning.

In a nutshell, this study proposes a tool that is capable of (a) using Big Data in terms of collecting and processing past observations/measurements of diverse sources; (b) building knowledge on the network load that is observed with respect to the past observations; (c) exploiting the knowledge obtained so as to predict the network load under predefined conditions; and thus, (d) offering insights into decision making mechanisms, which are responsible for dynamically managing network resources and guiding potential reconfigurations of the network.

5.3.2 The Tool

The proposed tool is divided into two phases consisting of one mechanism for each (Figure 5.4) and is tested on a network of Wi-Fi hotspots. During the first phase, the mechanism collects observations/data with respect to the observed network load, the Wi-Fi hotspot, the timestamp, the weather, and the selected feature of the day. One or more of the following parameters are used as input depending on the considered scenario: (i) the area/access point (AP0, AP7, AP8, AP37, AP64 or AP66) expressed in a 6D variable that consists of 0s and 1s depending on the access point from which the observation was received, e.g., 100000 for AP0 or 001000 for AP8; (ii) the time expressed in minutes (0-1440 minutes); (iii) the day (Sunday to Saturday) expressed in a 7D variable that consists of 0s and 1s (similarly to the access points, e.g., 1000000 for Sunday or 0001000 for Wednesday); (iv) the week of the year expressed as an integer (1-52); (v) the mean temperature of that day in Celsius; (vi) the precipitation of that day in millimeters; (vii) (bank) holidays expressed in a boolean way (0 or 1); and (viii) the observed load from the access point to the end users under these circumstances in Mbps. These data are then processed/ combined so as to transform to information of interest for the NO, i.e., to knowledge, with respect to the pattern that the load exhibits at a given area in the period of time and/or to the environmental conditions and/or the (bank) holidays.

Given the self-similarity nature of network traffic, i.e., the fact that the pattern (when and how the load increases and/or decreases) slightly changes, learning the pattern is expected to offer insights into the load to be encountered in the future (near or distant) at this area. In particular, this knowledge becomes available and is provided to the NO when the latter triggers the second mechanism, i.e., the proposed load prediction mechanism, through a request that specifies some or all of the following parameters (depending on which were used during the 1st phase): (i) the area/access point, (ii) the time, (iii) the day, (iv) the week of the year, (v) the expected mean temperature, (vi) the expected precipitation, and (vii) if the day of interest is going to be a holiday or not. Alternatively, the trigger could include only the area/access point, the time and the date, leaving the rest of the information to be completed by a preprocessing module that would receive them from e.g., a calendar RSS feed.

Consequently, the load prediction mechanism that uses the knowledge on the traffic pattern of all the areas with respect to the time and the considered parameters is able to predict the network load. These predictions are returned to the NO expressed in Mbps and followed by a percentage that designates the certainty of the mechanism with respect to this prediction, i.e., the probability of being correct. Sub-sections 5.3.2.1 and 5.3.2.2 present the two mechanisms of the tool, i.e.,

a) the knowledge building and data minimization mechanism, which is capable of handling large amounts of data coming from long time-windows, i.e., multiple months and diverse sources and which derives useful and meaningful, higher level information regarding the load of the network; and

b) the learning-based mechanism for a load prediction mechanism that will predict the load in the future and provide the NOs with such an insight.

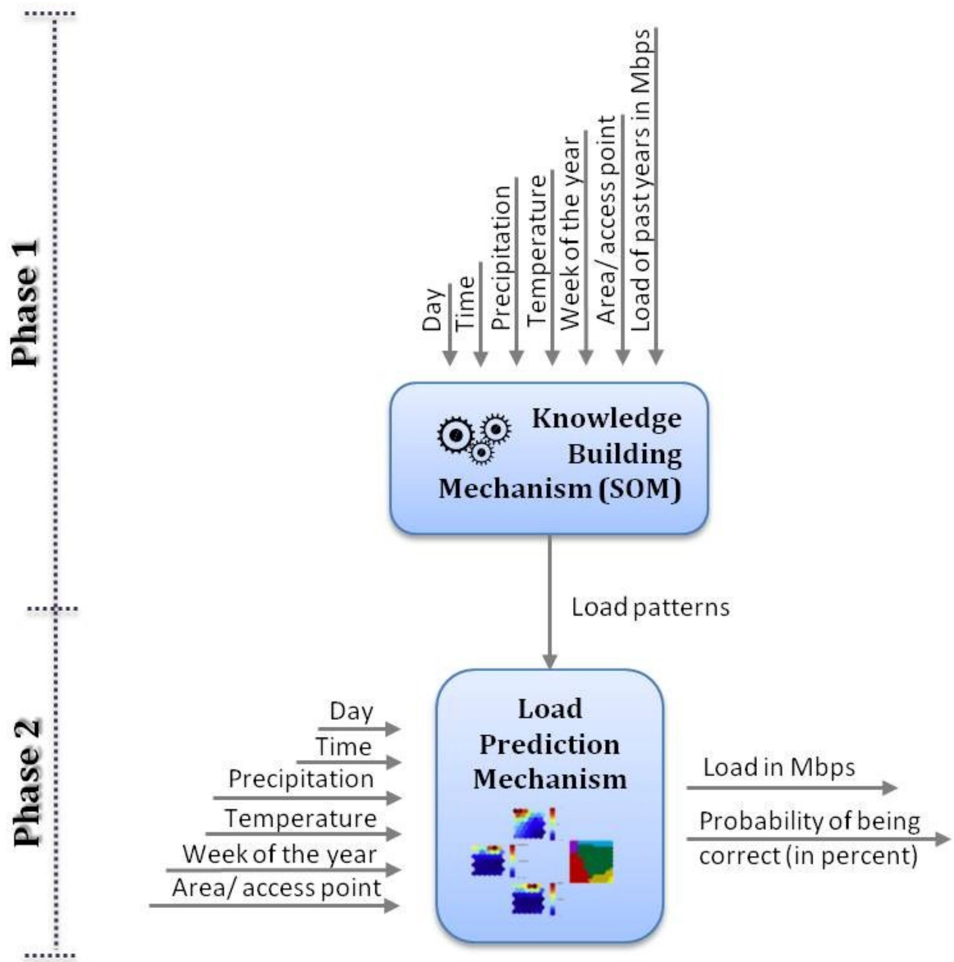


Figure 5.4: Phase 1 (upper part of the figure): Building knowledge on the pattern of the load; (b) Phase 2 (lower part of the figure): Use of the built knowledge for predicting the load.

5.3.2.1 Knowledge Building and Data Minimization

In order to build knowledge from the past experience of the network and not only from its current state, this mechanism needs to include a machine learning technique. The selected technique in this study is SOM (see sub-section 3.4).

In the context of this study, the dimensions of the data reach up to 19 (depending on the scenario) and comprise 6 dimensions for the area/access point, 1 for the time, 7 for the day, 1 for the week, 1 for the mean temperature, 1 for the precipitation, 1 for holidays and 1 for the load. The analysis of these dimensions for scenarios 2 and 3 is summarized in Table 5.2 while the respective dimensions for scenario 1 are depicted in Table 5.3.

It should be noted here that, although the total number of parameters that were used for each data sample is 19, only 18 of them were used for the creation of the clusters and the knowledge building, i.e. the components of the vectors were only 18. The measurements of the load (19th parameter/dimension) have been incorporated in the data sample only to be used later during the load prediction.

The mechanism proposed in this sub-section is based on a hybrid SOM, namely the PLGSOM as defined in [1], i.e., a version that is capable of adjusting both its size and its learning parameters. This occurs in order to enhance its performance and to allow the use of the technique without requiring a priori knowledge of the data set that will be used and/or human intervention to test and specify the optimal value of the algorithmic parameters. The less the parameters that need to be tuned, the more self-adaptable the technique. In this context, a parameterless growing SOM is more appropriate for online learning mechanisms in self-adaptive/autonomic systems, like the ones envisaged for future networks. Here, it is used for building the knowledge that is represented by a map similar to the one in the final phase of Figure 5.4.

In addition to this, and in order to verify the improved performance of the PLGSOM compared to the parameterless SOM, their performances were tested using the same data sets and comparing one to the other. The performance metric that was used is the MSE. In an indicative test case, the MSE when using the PLGSOM was equal to 0.000553 while in the case of the parameterless, the MSE was equal to 0.000563. This shows that the PLGSOM performs even better than parameterless SOM.

Table 5.2: Summary of the input variables, their dimensions, and the parameters

Variable	***Dimensions***	***Parameters***
Access Point	6	AP0, AP7, AP8, AP37, AP64, AP66
Time	1	T
Day	7	Sun, Mon, Tues, Wed, Thur, Fr, Sat
Week of the year	1	WoY
Mean Temperature	1	MT
Precipitation	1	P
Holiday	1	H
Load	1	L

5.3.2.2 Learning-based Mechanism for Load Prediction

As explained in Sub-section 5.3.2.1, the SOM technique is used to build the knowledge on the load that is observed with respect to the geographical area, the time, the day, the week, the mean temperature, the precipitation, and the (bank) holidays. The load prediction is made based on this knowledge. In particular, the prediction is based on the fact that when mapping new data samples on the designed map, the new data sample is expected to be mapped among similar ones. Thus, knowing the load that was observed with those data samples, one can safely predict the load that is related to the new data sample. More precisely, the predicted value of the load is considered as equal to the mean value of the loads that have been mapped on the specific cell where the new data sample was mapped. In order for this to be accomplished, the load prediction requests should be in the form of a message consisting of similar input parameters to those used during the training of the map.

Moreover, the load prediction is followed by a percentage C that designates the certainty of the mechanism with respect to this prediction, i.e., the probability of being correct. This percentage is calculated by (5.1), where σ and ΔL are the standard deviation and the range of the loads that have been mapped on the cell on which the data sample in question was mapped, respectively.

$$C = (1 - \frac{\sigma}{\Delta L/2}) \tag{5.1}$$

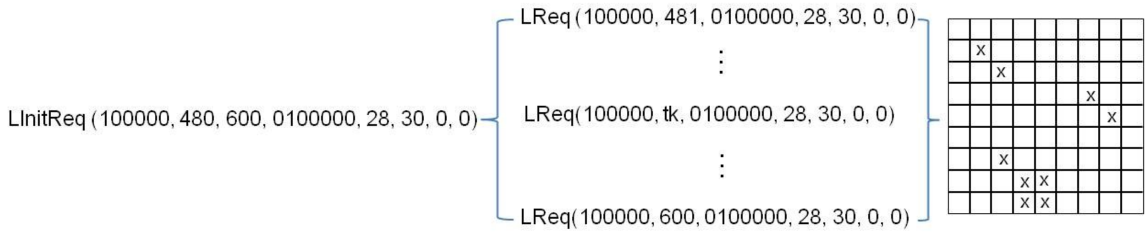
This percentage can also be used as a trust metric, i.e., as a metric of moving forward with or without this prediction. For example, if a network operator received an answer of the form (10, 99%) that would inform him that the load for this context will be 10 Mbps with a certainty of 99%, then he would most probably move forward with this prediction. On the other hand, if he received an answer like (100000, 40%), which informs him that the mechanism predicts a 100Gbps load but is only 40% sure of this prediction, then the network operator probably would decide to avoid any reconfiguration as he is not that sure of this prediction.

Additionally, the mechanism is capable of providing insight with respect to the future load of an area either in a time period equal to the observation interval time or for longer time periods by calculating the average of the network load for them. In other words, the request includes the following parameters: (i) the area/access point in question AP; (ii) the time period in question (T0 - first minute of the period and Tf – last minute of the period); (iii) the day in question- DayID; (iv) the week of the year in question- WoY, e.g., 28th week of the year; (v) the expected temperature of that day - T; (vi) the expected precipitation – P; and (vii) if it is going to be a holiday or not - H. Thus, the request will be formed as LInitReq(AP, T0, Tf, DayID, WoY, T, P, H).

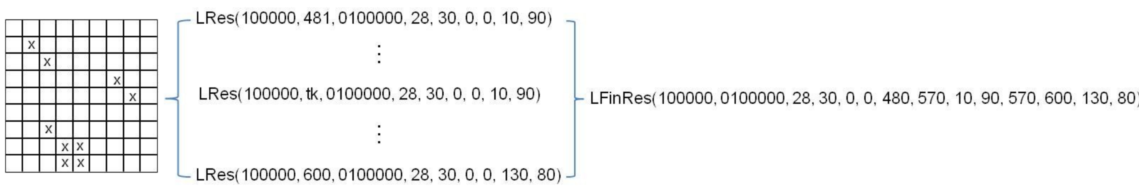
This request (from now on called initial request) is then broken into more requests, equal to the number of times that the time interval of the observations fits in the requested period. For example, if the period is 1 hour and the time interval is 15 minutes, then the initial request is split in 4 new requests. These requests, consisting of (i) the area/access point in question- AP, (ii) the time in question- tk, (iii) the day in question- DayID, (iv) the week of the year in question- WoY, (v) the expected temperature of that day - T, (vi) the expected precipitation – P, and (vii) if it is going to be a holiday or not - H are formed as LReq(AP, tk, DayID, WoY, T, P, H), and are mapped on the created SOM. The load of each request is equal to the mean value of the loads that had been mapped on the specific cell during the knowledge building phase.

Thus, the SOM will return the load prediction L_n in Mbps and the probability of being correct is the mechanism of this prediction in percentage C_n through equal (in the number) responses of the form LRes(AP, t_k, DayID, WoY, T, P, H, L_n, P_n). Finally, the mechanism responds to the request by sending back an aggregated response that consists of (i) the area/access point - AP; (ii) the day - DayID; (iii) the week of the year - WoY; (iv) the expected temperature of that day - T; (v) the expected precipitation - P; (vi) if it is going to be a holiday or not - H; (v) the first minute $T_{1,0}$ and the last minute $T_{1,f}$ of the 1st sub-period of the time period in question which is comprised of minutes with an equal load; (vi) the load prediction of this sub-period L_1; (vii) how certain the mechanism is of this prediction C_1; (viii) the first minute $T_{2,0}$ and the last minute $T_{2,f}$ of the 2nd sub-period of the time period in question; (ix) the load prediction of this sub-period L_2; (x) how certain the mechanism is of this prediction C_2, etc., i.e., the final response is in the form of LFinRes(AP, DayID, WoY, T, P, H, $T_{1,0}$, $T_{1,f}$, L_1, C_1, $T_{2,0}$, $T_{2,f}$, L_2, C_2, ..., $T_{n,0}$, $T_{n,f}$, L_n, C_n).

The above described process with an indicative example is depicted in Figure 5.5. According to the example, let's assume that the initial request was "what will the load be at AP0 (100000) during the period that starts from the 480th minute of the day and ends at the 600th minute of the day, on Monday (0100000), the 28th week of the year given the fact that the temperature will be 30^{o}C, it won't rain, and will not be a holiday?". Thus, the request would be LInitReq(100000, 480, 600, 0100000, 28, 30, 0, 0). The mechanism will split this initial request in 8 new requests, one for each time that the interval time (15 minutes) fits in the time period of 120 minutes, i.e., will split the initial request to 8 new ones of the form LReq(100000, 480 < t_k < 600, 0100000, 28, 30, 0, 0) and map them. For each such request, the SOM will send a response similar to LRes(100000, 480 < t_k < 600, 0100000, 28, 30, 0, 0, 10, 90). Finally, assuming that the load level prediction will be 10 Mbps for the period 480-570 with a certainty of 90% and 130 Mbps for the period 570-600 with a certainty of 80%, the mechanism will respond to the request with the following aggregated response: LFinRes(100000, 0100000, 28, 30, 0, 0, 480, 570, 10, 90, 570, 600, 130, 80).



(a)



(b)

Figure 5.5: Load Prediction Mechanism: An example: (a) a long request is treated as many "15-minute" requests, which are then (b) aggregated in one "Final" Response.

5.3.3 Scenario 1: Prediction of Network Traffic based on Time, Day, Location and Weather Conditions

In this scenario, the data that are used for the training of the map comprise i) the timestamp (in k-th minute of the day in UTC time zone), ii) the day (Monday, Tuesday, etc.) described in a binary format of zeros and aces e.g., 1000000 for Monday, 0100000 for Tuesday, etc., iii) the type of the area (e.g. residential or social areas) in a binary format as well, iv) the temperature of the area during that day in Fahrenheit, v) the precipitation of the area during that day in inches and vi) the network load that was observed in this area during the specific day and time in the past. These 5 variables are described by 17 components in total (1 for the timestamp, 7 for the day, 6 for the type of the area, 1 for the temperature, 1 for the precipitation and 1 for the network load) which consist the 17 dimensions of the data.

5.3.3.1 Data and Data Pre-processing

Part of the data file that was used for validating the mechanism was retrieved by CRAWDAD database [2]. In particular, the retrieved by [2] dataset refers to data collected in the Dartmouth College every 5 minutes from over 450 access points that are used from several thousand users of the college [3]. The exact trace can be downloaded by [4] and reflects the use of the access points during fall 2003 and 2004, i.e., from November 1st, 2003 up to February 29th, 2004. From this trace, the timestamp, the incoming Bytes and the type of the building that hosted the access point were exploited.

In order to complement the data file with information related to the environmental circumstances, data were also retrieved from [5]. Using a custom Java tool, the mean temperature and the precipitation of each day from November 1st, 2003 up to February 29th, 2004 for the area of Hanover, New Hampshire, i.e., the area where the Dartmouth College is located, were extracted.

The information retrieved from the two aforementioned sources was combined, pre-processed and eventually fed to the mechanism for training the SOM and building the respective knowledge. Moreover, subset or the entire data file was also used during the validation of the mechanism, i.e., during the prediction phase of the mechanism, depending on the type of the test that was performed each time (see also sub-section 5.3.3.2).

Before using the retrieved data and in order to create the needed data files a pre-processing phase took place. This pre-processing phase targets at i) transforming the information in a form recognizable and exploitable by SOM and ii) normalizing the data.

Since SOM recognizes and exploits vectors, data needed to be transformed accordingly. Towards this direction, data were grouped in data samples, each of which contained 1 value for each variable. Moreover, each variable could be described by one or more components. In particular, the description of the variables can be found in Table 5.3.

Table 5.3: Variables: their Components and their Units

Variable	***Components***	***Ubits/Potential Values***
Time	1: minute of the day	Net number from 0-1440, UTC time zone
Day	7: mon, tues, wed, thur, frid, sat, sund	Boolean (0/1), All the components are equal to 0 apart from the component that corresponds to the day which is equal to 1, e.g. if it is Tuesday then all the components are equal to 0 apart from "tues" which is equal to 1, i.e., the variable has the value 0100000. All the potential values are: 1000000 for Monday 0100000 for Tuesday 0010000 for Wednesday 0001000 for Thursday 0000100 for Friday 0000010 for Saturday 0000001 for Sunday
Location	6: Adm, Athl, Lib, Acad, Res Soc	Boolean (0/1), All the components are equal to 0 apart from the component that corresponds to the type of the building that hosts the access point which is equal to 1, e.g. if the access point is hosted in a residential building of the campus then all the

		components are equal to 0 apart from "res" which is equal to 1, i.e., the variable has the value 000010. All the potential values are: 100000 for Administrative buildings 010000 for Athletic buildings 001000 for Libraries 000100 for Academic buildings 000010 for Residential buildings 000001 for Social buildings
Temperature	1: temperature	Fahreneit
Precipitation	1: precipitation	Inches
network load	1: network load	bps

As it can be seen in the same table, the variables that needed more than one component in order to be described were variables that cannot be numerically organized or ranked. Thus, they needed an ID instead. On the other hand, numerical IDs from 1 to 7 could not be used either since this would imply e.g. that Monday's distance from Thursday is of higher importance than the distance between Monday and Tuesday, i.e., Monday differs more from Thursday than from Tuesday or that Friday's offered load pattern is more similar to Saturday's than it is to Monday's, which is something we cannot assume nor can we assume that there is a linear relationship between days of the week (1 through 7) and the observed network load. An efficient way of overcoming this issue was to describe such variables using more than one boolean components. This way, distances between e.g. the days are equal to each other (equal to $\sqrt{2}$).

Finally, before closing the pre-processing phase of the data, data had to be normalized. In particular, each of the components was normalized in [0, 1].

5.3.3.2 Results

This mechanism is proposed in the context of both predicting network load and of contributing to "Big Data" management, thus metrics that address both aspects have been considered. Regarding the 1st aspect, one comparative diagram for each type of buildings is provided. However, in order to have more quantitative metrics as well, MSE, Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) have also been used to showcase how close the predicted and the real values of the load are.

Moreover, two types of tests were performed. The first one used the same data samples for both training and validating the mechanism. In other words, the data samples that were used during the knowledge building phase and the load prediction phase of the mechanism were the same. This type of tests provided a validation of the mechanism to seen/ known data samples, i.e., to seen/ known circumstances.

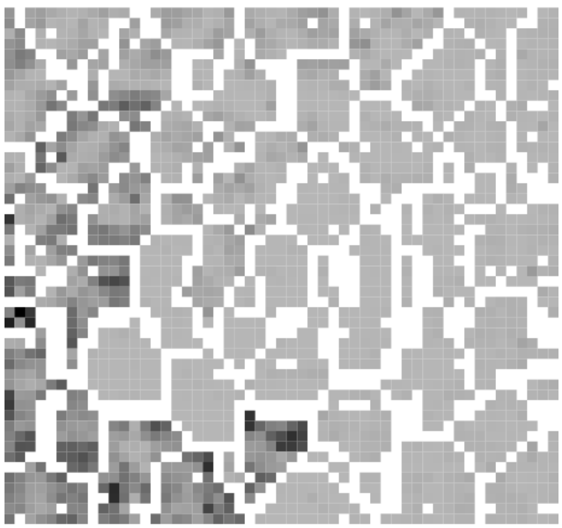
The 2nd type of tests, targeted at providing validation of the mechanism to unknown or at least not completely known situations. Thus, the tests involved different data samples during the knowledge building and the prediction phase of the mechanism. This type of tests is of even higher interest since they validate the mechanism in a context closer to the real working environment.

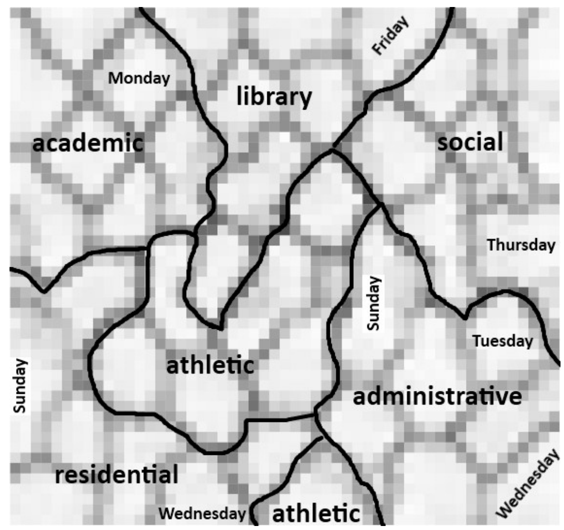
Regarding the 2nd aspect, i.e., the contribution of the mechanism to the "Big Data" management, indications with respect to how the decrease of the volume of data and thus decrease of the required storage has been accomplished are presented.

5.3.3.2.1 Knowledge Building – SOMs

The first issue that the mechanism deals with is knowledge building. The knowledge that is built is about the load that has been observed in the past with respect to the time, the day, the building of the campus, the mean temperature and the precipitation of the area during each day. Towards this direction, the mechanism exploits the SOM learning technique which depicts the data samples on a 2D map. Examples of such maps are depicted in Figure 5.6 and Figure 5.7.

In particular, Figure 5.6(a) depicts the map where one could observe the variations of network load: the darker the area of the map, the higher the load that had been observed under the context that is described from the mapped data samples. On this map it is also easy for someone to identify the separate groups of data, i.e. the different clusters. On the other hand, Figure 5.6(b) depicts a map that in known as U-matrix. A U-matrix presents only the clusters that have been created without focusing on any of the variables of the data samples. Using component maps, i.e. maps that focus on particular variables of the data samples, information about the type of the building that hosted the access points and exemplary information about the days has been added on top of the U-matrix.





(a) **(b)**

Figure 5.6: (a) SOM colored with respect to the network load: the darker the area, the higher the network load that has been observed in the past under the specific context that corresponds to the mapped data samples. (b) Created clusters on top of a U-matrix.

Although visualization of the maps per se is very interesting, the added value of the maps is in their combining for obtaining valuable information. Observing Figure 5.6(a) and Figure 5.6(b) in parallel one will notice that high load is monitored in specific types of buildings and not in all of them. More specifically, high load can mainly be seen in the residential and academic buildings. Enough load is also served by access points in libraries but still not that much as in the first two types of buildings. On the other hand, network load in athletic, administrative and social buildings is low when compared to the first three types of buildings.

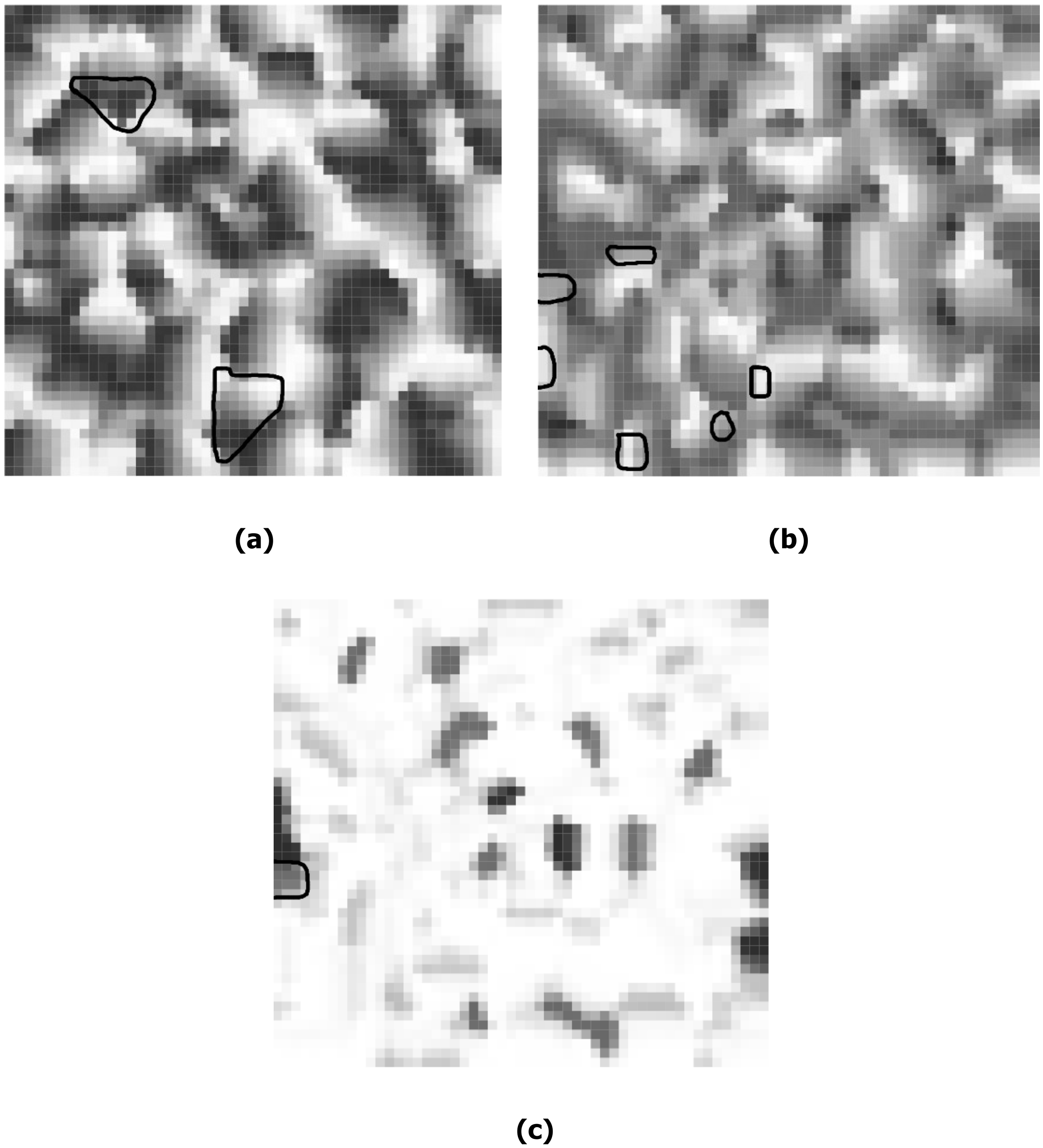


Figure 5.7: Component maps for (a) the time of the day, (b) the temperature and (c) the precipitation.

Component maps focusing on the time of the day, the mean temperature and the precipitation can be found in Figure 5.7. In particular, Figure 5.7(a) presents the component map that is related to the time of the day. The darker the cells the more advanced the time in the day (closer to midnight in the UTC time zone). Combining the information that is hold in Figure 5.6(a), Figure 5.6(b) and Figure 5.7(a) someone can make interesting remarks with respect to when the network is more loaded in a given type of buildings. For example, focusing on the areas that are highlighted in Figure 5.7(a), highly loaded time zone for the residential buildings starts in the late afternoon (in UTC time zone) and stops in the early morning of the next day (in UTC time zone). Accordingly, for academic buildings the more loaded time zone is from the early afternoon (in UTC time zone) until the end of the day (in UTC time zone). Transforming these observations in the time zone of Hanover, New Hampshire would result in the fact that highly loaded time zones are a) late afternoon until around midnight for the residential buildings and b) midday until the afternoon for the academic buildings.

Moving to Figure 5.7(b) and Figure 5.7(c), the temperature and the precipitation of the day are presented there. Higher temperatures or precipitation are mapped in darker colored cells of the respective figures. Similarly to the above described use of the maps, these maps, when combined with the maps of Figure 5.6, can correlate the temperature or the precipitation of a given day with the load of a specific type of buildings. Indicative examples for these cases are the circled areas of Figure 5.7(b) and Figure 5.7(c) respectively. Given the fact that the building type is residential, there is indication that mean temperature and precipitation affect network usage as depicted in Figure 5.7(b) and Figure 5.7(c), the marked areas of which denote low temperatures or high precipitation, the same areas in Figure 5.6(b) denote a residential type of buildings and finally the same areas in Figure 5.6(a) indicate high load. On the contrary, as expected and depicted in the figures, network usage of the administrative or the athletic buildings is not affected by the temperature or the precipitation. Thus, low temperatures or high precipitation corresponds to high load in the network in the residential area but not in administrative and athletic ones.

Similar to the above insights, but more numeric ones, can be provided by that part of the load prediction mechanism which is responsible for the prediction per se.

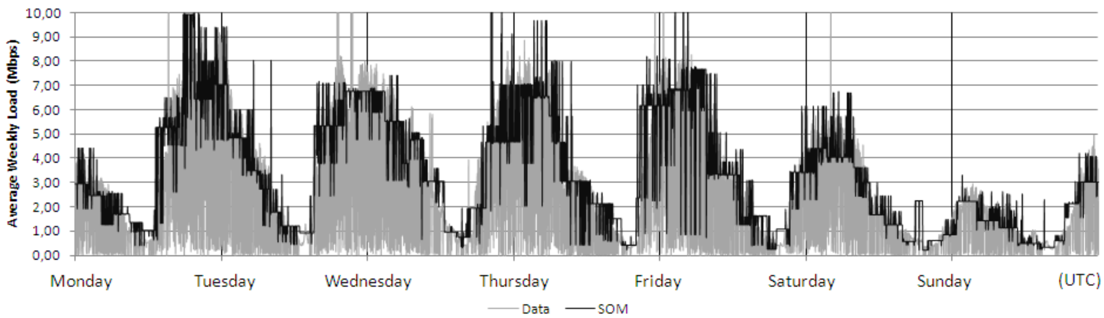
5.3.3.2.2 Knowledge Exploitation: Load Prediction

Although the above presented remarks can be used as patterns that the load follows, they are quite general. More specific information with respect to network load can be provided by the load prediction mechanism which actually exploits the maps and is capable of processing them in more details than human eye.

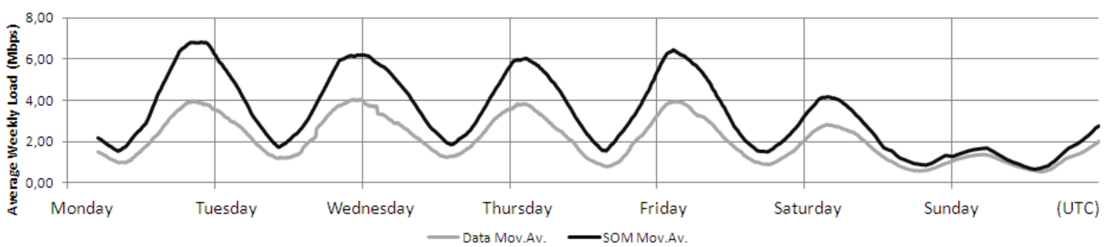
For validating the part of the mechanism that was related to the prediction of the load per se comparative diagrams, MSE, RMSE and MAE were used. Moreover, the performance of the mechanism was examined in tests of 2 types: a) tests where the trained map had seen the exact same data samples (i.e. the exact same context had been encountered in the past) and b) tests with unseen data, i.e. with data samples that had not been used during the training of the map.

To begin with, the first tests that took place used the same data samples that were used for the training of the map, i.e., the training data samples. For each data sample, a load prediction was requested. Moreover, the real value of the load of these data samples was known from the real measurements. The average values of both the predicted and the measured load in a time period of a week were calculated and their diagrams were drawn resulting in figures like the one depicted in Figure 5.8(a). However, they were quite difficult to be compared. In order to resolve this issue, centered moving averages over 200 data samples were calculated for both the predicted and the real values of load and the respective diagrams were drawn as well (Figure 5.8(b)).

Figure 5.9 depicts those diagrams for all the types of buildings, namely the a) academic, b) residential, c) libraries, d) social, e) administrative and f) athletic ones. Moreover, the diagrams have been depicted in pairs (load prediction and real measurements) for facilitating their comparison. As it can be observed by these diagrams, although the predictions are not fully in line with the real measurements, SOM has adequately learnt the pattern of the load and the mechanism is capable of predicting it with small deviations. The above observation is also supported by the values of the MSE, the RMSE and the MAE which are equal to 0.001012, 0.031819 and 0.015416 (when using the normalized values of load) respectively. Thus, the predicted values of load deviate on average only ±1.174 Mbps from the real measured ones. Even these deviations do not necessarily come from faulty predictions. Such deviations often come from the bursty nature of internet traffic, i.e. from the fact that although the internet traffic may follow some patterns, there are still some bursts that do not follow the patterns. Those bursts cannot be predicted and thus increase the error metrics. On the other hand, being capable of predicting those bursts would imply that SOM has learnt very well the pattern of the specific data, i.e. has been overtrained, and thus is unable of generalizing and predicting network load in unseen data.



(a)

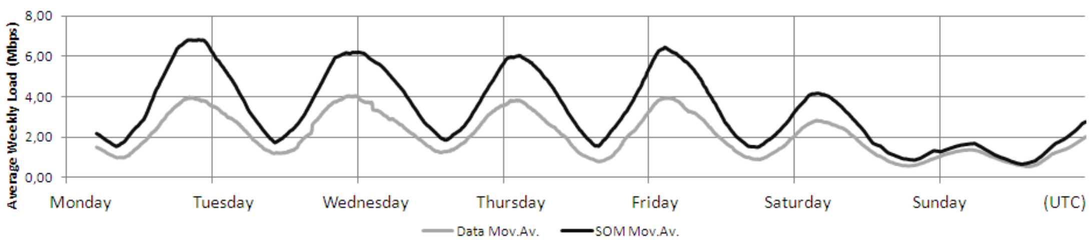


(b)

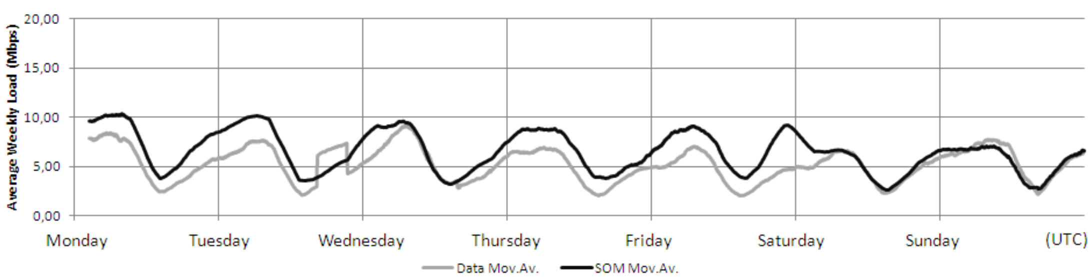
Figure 5.8: Comparative diagrams: (a) load predictions and real measurements, (b) moving average of the load predictions and the real measurements

In order to validate how well SOM can generalize, the performance of the mechanism was also tested in unseen data, i.e., data that had not been used during the training of SOM. Towards this direction, the data file was randomly split in two data files, one containing the 70% of the available data samples and the other one containing the rest 30% of the available data samples. The 1st data file was then used to train the map. The 2nd one was used for evaluating the performance of the mechanism.

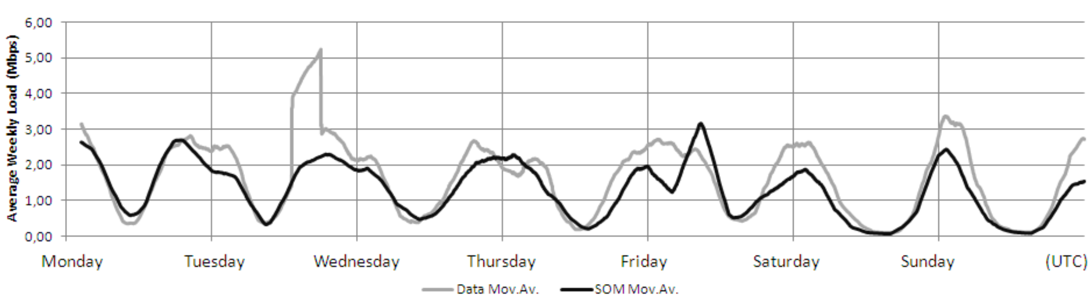
Following the same process as before, the data samples of the 2nd data file were inserted in the mechanism as triggers for which the load had to be predicted. Both the real values (coming from the data file) and the predictions were averaged in one week's time frame and their moving averages over 200 data samples were calculated. Eventually, the diagrams of the moving averages of both the real and the predicted values were drawn and are depicted in Figure 5.10. Similarly to the case when "seen" data were tested, the diagrams are presented in pairs of two (predicted and real values of each building type) for facilitating the comparisons.



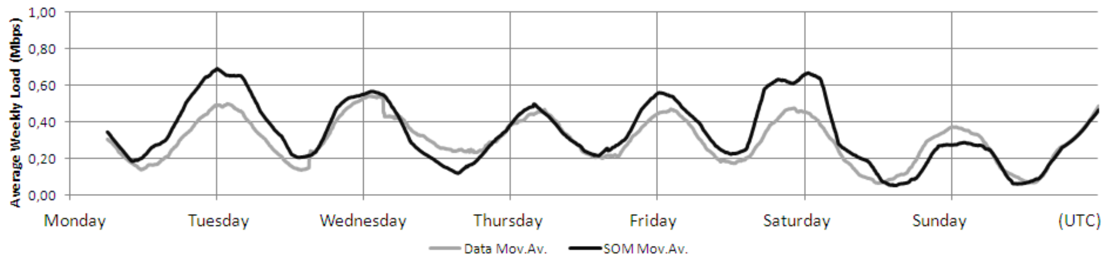
(a)



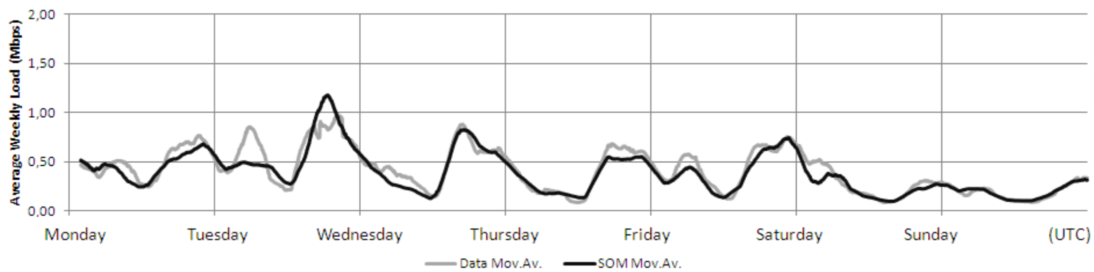
(b)



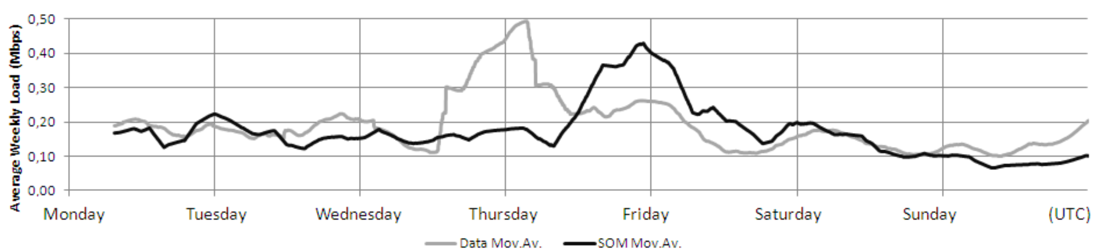
(c)



(d)



(e)

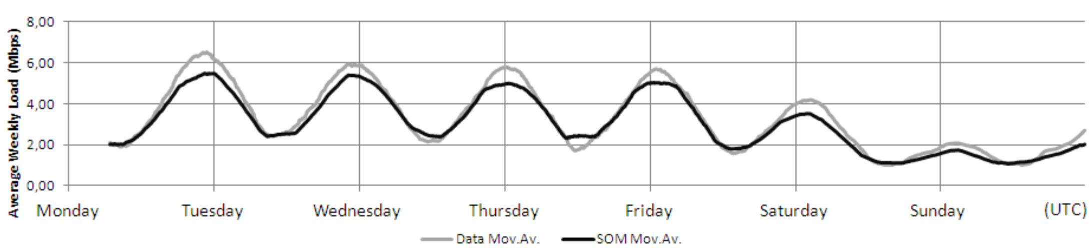


(f)

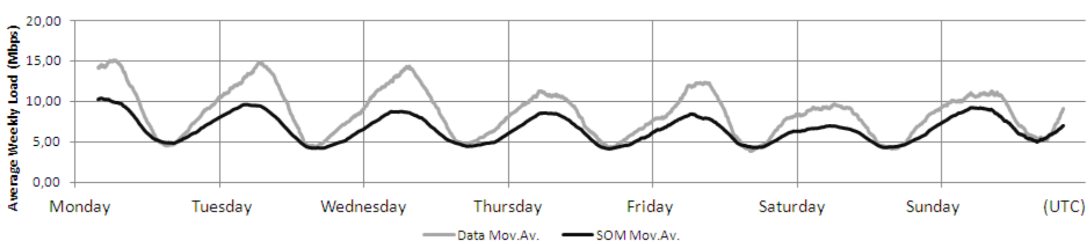
Figure 5.9: Comparative diagrams of the moving average of the load predictions and the real measurements in (a) Academic buildings, (b) Residential buildings, (c) Libraries, (d) Social buildings, (e) Administrative buildings and (f) Athletic buildings with "seen" data.

Comparing these diagrams and the values of the MSE, the RMSE and the MAE which are equal to 0.001168, 0.034176 and 0.016981 (in the normalized values of load) respectively confirm that the mechanism can predict the load of even unknown or at least slightly different from those encountered by the network in the past data samples. In fact, the predicted values of such cases deviate on average from the real ones only ±1.293 Mbps.

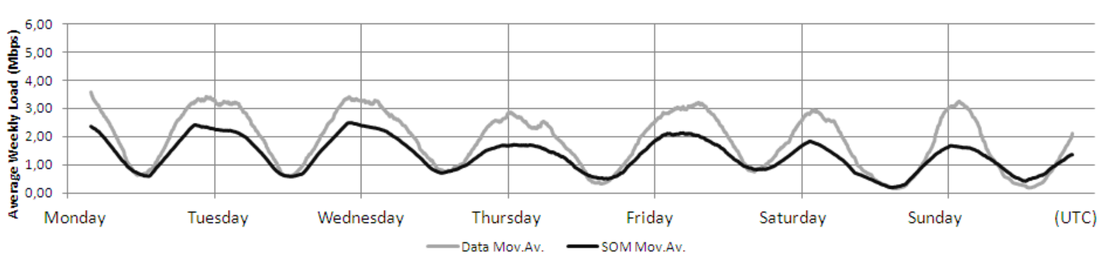
Finally, the performed tests, using either "seen" or unseen data, also showcased some interesting tradeoffs. Indicative examples on which the following analysis will be based can be found in Table 5.4. First of all, comparing the 1st and the 2nd line of the table one can observe that using unseen data slightly deteriorates the performance of the mechanism. Moreover, comparing the 6th with the 7th and the 1st, the 3rd and the 4th lines to each other, a tradeoff between the size of the map, i.e. the storage requirements of the mechanism, and the performance of the mechanism is also observed. In fact, larger maps demonstrate slightly better performance.



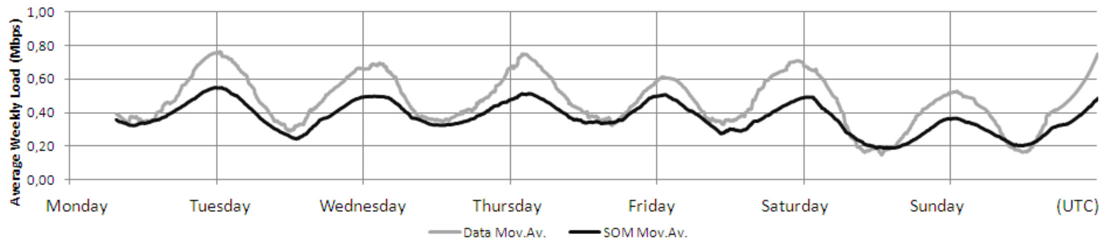
(a)



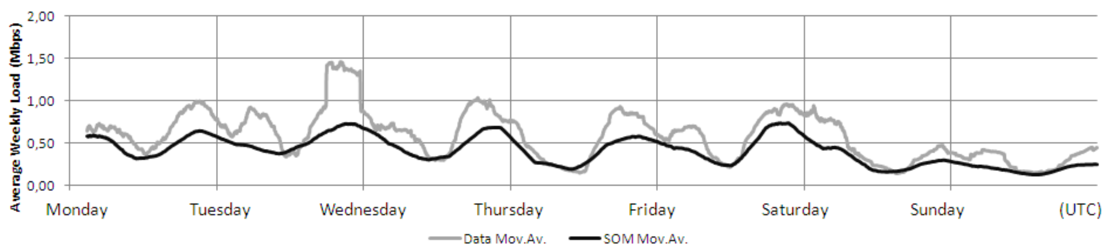
(b)



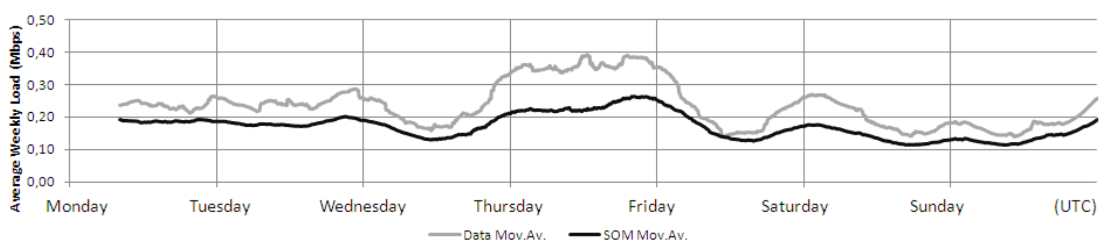
(c)



(d)



(e)



(f)

Figure 5.10: Comparative diagrams of the moving average of the load predictions and the real measurements in (a) Academic buildings, (b) Residential buildings, (c) Libraries, (d) Social buildings, (e) Administrative buildings and (f) Athletic buildings with "unseen" data.

The last observation is related to how immune the mechanism is to rare and excessively high values of load. In order to do so, the around 15 exceptional data samples the load value of which exceeded 76,146,848 bps, i.e. the upper bound of load values for most data samples, and reached up to 437,142,176 bps were changed. More specifically, in order to fit to this upper bound, their load value was substituted by 76,146,848 bps (clamped data). However, comparing the results of such tests with tests where no data had been clamped revealed that clamping data does not significantly improve the performance of the mechanism. On the contrary, MSE and MAE of clamped data become worse (due to the smaller normalization range equal to 76,146,848) while their denormalized MAE does not seem to be notably lower than the one observed in cases where data have not been clamped. In other words, the mechanism is immune to the rare and excessively high values of load.

Table 5.4: Indicative Examples of Performed Tests

Clamped Data	***Data Samples***	***Map Size***	***Unseen Data***	***MSE***	***MAE***	***De-normalized MAE (bps)***
Y	154,514	25x36	Y	0.00117	0.01698	1,239,049
Y	141,514	25x36	N	0.00115	0.01685	1,282,693
Y	141,514	42x55	Y	0.00105	0.01573	1,198,094
Y	141,514	30x30	Y	0.00118	0.01710	1,302,415
N	141,514	41x55	Y	0.0000	0.00276	1,204,764
Y	202,163	50x53	N	0.00101	0.01542	1,173,879
Y	202,163	45x62	N	0.00100	0.01522	1,158,726
N	202,163	57x46	N	0.00004	0.00270	1,179,847

5.3.3.2.3 Contributing to "Big Data" management

Although the main focus of this research is on developing a mechanism capable of predicting network load given the time, the day, the area, the temperature and the precipitation, aspects related to the Big Data phenomenon have also been addressed. More specifically 2 aspects related to Big Data have been addressed, namely a) how Big Data can be used for providing insights and b) volume reduction of Big Data.

To begin with, for the 1st aspect we consider Big Data from their diversity point of view. Towards this direction, the study presents a mechanism that is based on machine learning and exploits data which come from different to each other sources in order to predict future behaviors of the network. The proposed mechanism is generic enough to be reusable in other problem statements as well.

Towards the 2nd aspect, Big Data have primarily been considered from the volume point of view. The diverse unstructured data have been grouped in data samples, i.e., structured data of multiple dimensions. Specifically, 3,436,771 unstructured data have been grouped in 202,163 data samples of 17 dimensions. The latter, i.e. the 202,163 17D data samples, were then mapped on a 2D map further reducing the volume of the data from 202,163 17D data samples to 202,163 2D vectors, i.e. decreasing their dimensionality. Eventually, from the 202,163 2D vectors only some representative of them needed to be stored. Those were equal to the number of cells of the created map, i.e. not more than 2,650 vectors and not less than 900 vectors. Overall, the mechanism managed to reduce the data at least from 3,436,771 unstructured data to 2,650 vectors or at most from 3,436,771 unstructured data to 900 vectors.

5.3.4 Scenario 2: Prediction of Network Traffic based on Time, Day, Location, Week of the Year and Bank Holidays

In this scenario, the observations will comprise information with respect to i) the area/ access point (AP0, AP7, AP8, AP37, AP64 or AP66) expressed in a 6D variable which consists of 0s and 1s depending on the access point from which the observation was received (e.g., 100000 for AP0 or 001000 for AP8), ii) the time expressed in minutes (0-1440 minutes), iii) the day (Sunday to Saturday) expressed in a 7D variable which consists of 0s and 1s (similarly to the access points, e.g. 1000000 for Sunday or 0001000 for Wednesday), iv) the week of the year expressed as an integer (1-52), v) holidays expressed in a boolean way (0 or 1) and vi) the observed load in Mbps under these circumstances. Accordingly, after building the knowledge, the mechanism will be capable to predict future load in Mbps given i) the area/ access point, ii) the time, iii) the day, iv) the week of the year and v) if the day of interest is going to be a holiday or not.

5.3.4.1 Data and Data Pre-processing

The data that were used for the research were retrieved by CRAWDAD database [6]. In particular, the retrieved dataset refers to user session traces which were collected from a large number of free Wi-Fi hotspots of "Île sans fil" [7] in Montréal, Québec, Canada for three years. In particular, the used trace [8] contains 587,782 user sessions for 69,689 (distinct) users, which were collected from 206 hotspots for the time period from August 28th, 2004 up to August 28th, 2007.

From this trace, the timestamp, the incoming Bytes and the access point ID of the entries that were related to the 6 access points with the highest load were exploited. Namely, those access points were the AP0, AP7, AP8, AP37, AP64 and AP66. These data were then pre-processed so as a) the timestamp to be translated in the minute of the day, the day (expressed as a 7D variable) and the week on the year, b) the interval between the entries to be equal to 15 minutes and c) the access point ID to become a 6D variable. Let us note here that similarly to scenario 1 (sub-section 5.3.3) both the day and the access point IDs have to be expressed as 7D and 6D variables respectively so as to be processed correctly by SOM.

Moreover, the data had to be complemented with the holidays. As soon as this was completed, for each load measurement there were data designating i) the access point from which the load measurement had been monitored, ii) the time, iii) the day and iv) the week of the year that the load was monitored and v) if that day was a holiday or not. The number of the dimensions of each variable and their respective parameters can be seen in Table 5.2. In total, the parameters that were used for this scenario for each data sample, and thus its dimensions as well, are 17 (out of the 19 depicted in the table). The 16 of them were used for the creation of the clusters and the knowledge building but the measurement of the load (17th parameter/ dimension) has been incorporated in the data sample only for being used latter during the prediction phase.

Eventually, the data had to become normalized and to be in a recognizable by the SOM format. The latter means that the data had to be organized in data samples, i.e., in groups each of which contained one value for each of the 6 variables (access point ID, time, day, week of the year, holiday and load measurement).

5.3.4.2 Results

Having developed the mechanism all that was left to be done was the evaluation of its performance. Towards this direction numerous of tests were performed providing both quantitative and qualitative results. Hereafter some indicative results are presented.

Similarly to scenario 1 (sub-section 5.3.3), the tests can be grouped in two types: a) those which used the same data for both building the knowledge and evaluating the mechanism and b) those which used different data for building the knowledge and evaluating the mechanism. The main difference among them is lying in the fact that in the first case the mechanism has "seen" the exact same data and thus has the specific experience while in the second case the mechanism is familiar with similar but not the same data, i.e., has similar experiences from which it will try to exclude the safest conclusions and predict the future load.

The methodology of the evaluation was as follows: Data samples were sent to the mechanism as queries asking for the load that will be encountered if their context applied. The mechanism provided its prediction for each one of them. These predictions were eventually compared to the really monitored values (found in the dataset) of the mechanism either in terms of MSE, RMSE and MAE for the quantitative results or as diagrams for the qualitative results. The three different options of errors were provided for facilitating future comparisons of the mechanism with other similar mechanisms since their variety increases if more metrics are available.

Finally, the last sub-section reveals how the mechanism contributes to some of the management issues of the Big Data. Numerical results towards this direction are provided as well.

5.3.4.2.1 Quantitative Results

Table 5.5 summarizes some indicative tests and their respective quantitative results. In the table, the data that were used for the knowledge building are referred to as training data while those used for the evaluation of the mechanism are named as evaluation data. Moreover, the table also points out the size of the map, measured in cells, which in the case of growing SOM also represents how adjusted the map is to the training data.

Moreover, for the purposes of the evaluation of the mechanism the initial dataset was split in three sub-datasets: a) Y1 which refers to the dates from August 28th, 2004 up to August 28th, 2005, b) Y2 involving data from August 28th, 2005 up to August 28th, 2006 and c) Y3 which expands from August 28th, 2006 up to August 28th, 2007. In addition, for each of the created maps (1-3) two tests have been performed, one using the training data for the evaluation as well (1a, 2a and 3a) and one having different training and evaluation data (1b, 2b and 3b).

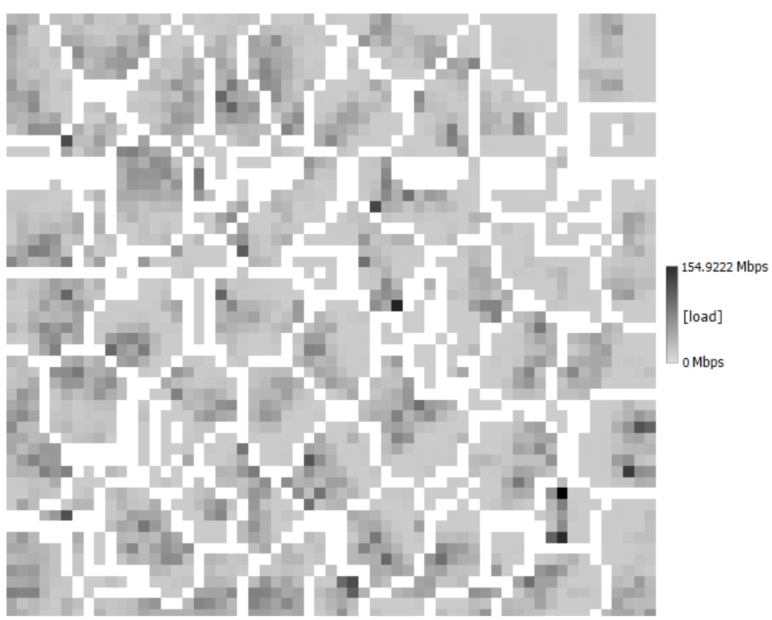
Comparing the results presented in the table, the first thing that is noticed is that the mechanism performs significantly better when the trigger refers to a case that has already been "seen" by the SOM. This observation was expected and that is also one more reason why online training has been considered.

Table 5.5: Quantitative Results

	Training Data	***Map Size***	***Evaluation Data***	***MSE***	***RMSE***	***MAE***
1a	Y1 & Y2	3245	Y1 & Y2	0.00018	0.01342	0.00541
1b	Y1 & Y2	3245	Y3	0.00055	0.02353	0.00783
2a	Y2	2650	Y2	0.00018	0.01342	0.00541
2b	Y2	2650	Y3	0.00058	0.02400	0.00958
3a	Y1 & Y2	900	Y1 & Y2	0.00019	0.01374	0.00564
3b	Y1 & Y2	900	Y3	0.00055	0.02345	0.00781

Another useful observation is the fact that when only Y2 was used for building the knowledge (cases 2a and 2b), although the performance on seen data remains the same (comparing to case 1a), the performance in unseen data slightly deteriorates. This test was selected mainly for identifying if observations older than one year deteriorate the performance of the mechanism. According to the obtained results it seems that older observations not only they don't deteriorate the performance of the mechanism but they even enhance it. This is probably due to the fact that less data have contributed to the knowledge building and thus SOM is familiar with fewer combinations of the involved variables. This affects the performance of the mechanism on the unseen dataset since the mechanism has even less past experience but not the performance on the seen dataset since the mechanism has already received all the experience it needs for predicting the load of these data samples.

The last observation, which comes from the comparison of 3a and 3b with 1a and 1b respectively, is that smaller maps result in better performance of the mechanism when the evaluation data differ from the training data and worse performance when the same data are used for both the knowledge building and the evaluation. This is related to the fact that the bigger the map is, the more adjusted to the training data it becomes. In machine learning, this is also known as overtraining which eventually results in the learning technique being unable to generalize well. For avoiding such cases, the performance of the technique between all the test cases (i.e., with seen or unseen data) should be kept as close as it gets. Thus, the best performance of the mechanism among the 3 presented in the table is considered to be the one using the 3rd map. This map is also depicted in Figure 5.11a while Figure 5.11b, Figure 5.11c, Figure 5.11d and Figure 5.11e depict the same map with information regarding the formulated clusters in terms of access points, holidays, days, time and week of the year.



(a)

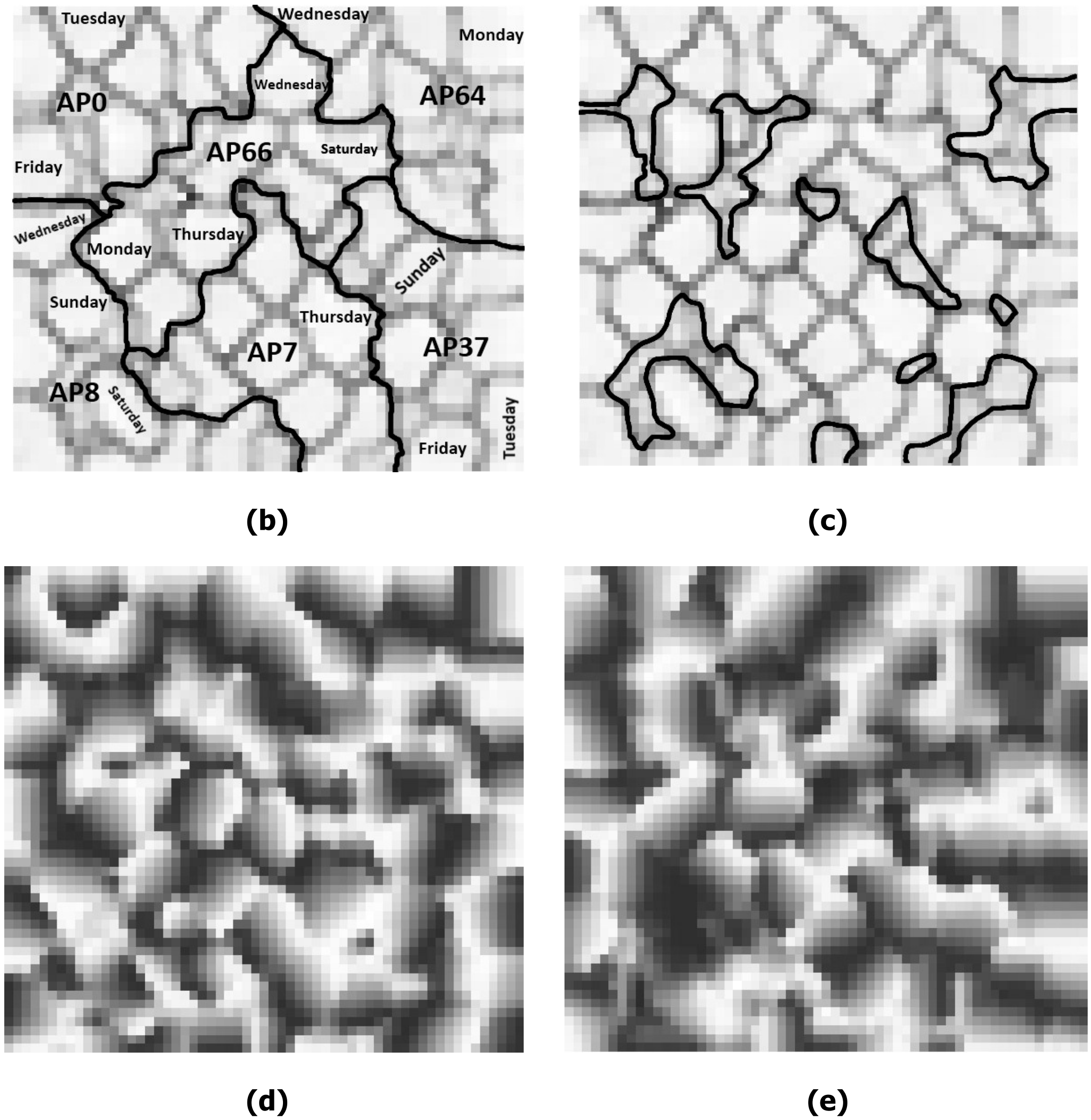


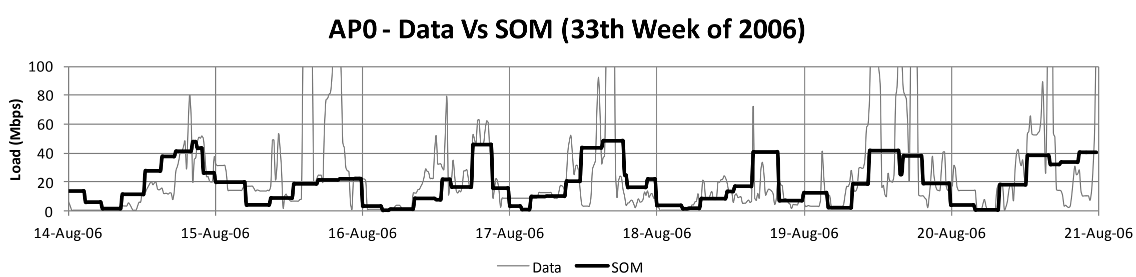
Figure 5.11: Map 3 from the (a) load, (b) access points and days, (c) holidays, (d) time and (e) week of the year points of view. The marked areas in the (c) refer to the holiday clusters while in (d) and (e) the higher the value of the respective parameter, the darker the area on which the data sample is mapped.

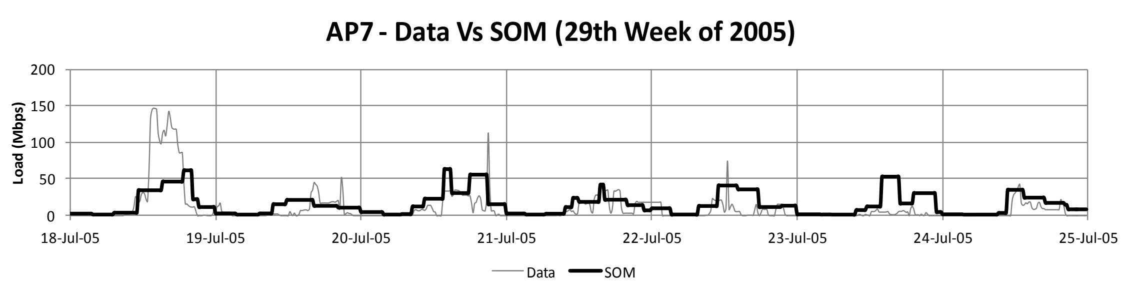
5.3.4.2.2 Qualitative Results

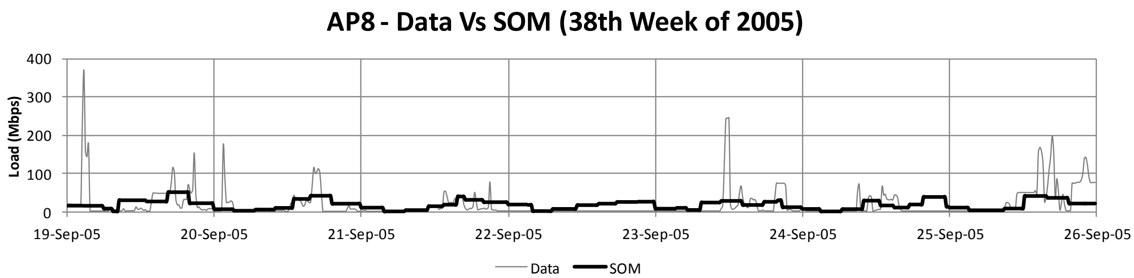
The qualitative results refer to comparative diagrams between the predicted and the real load values. Figure 5.12 and Figure 5.13 depict examples for each access point for seen and unseen evaluation data, respectively.

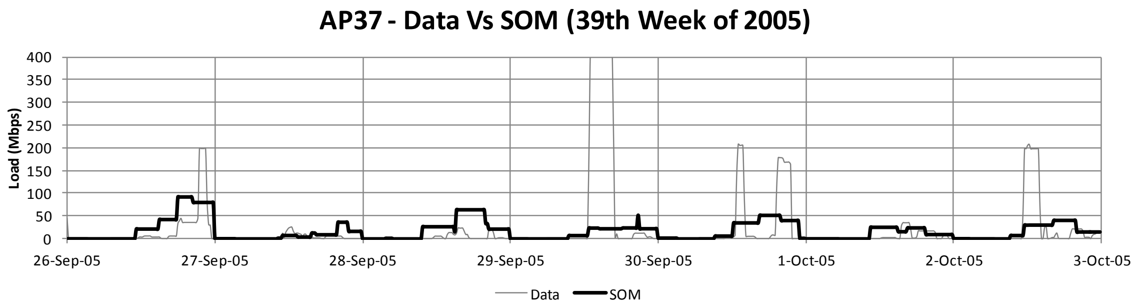
As can be observed from the comparative diagrams of both figures, although the mechanism has learnt the pattern of the load, i.e. when there is a load, it fails to predict some of its peaks. This may be caused by unpredictable events that suddenly attract (with no prior similar experience) more users, e.g., the place where the access point is hosted became more popular for some reason, or the access point next to it stopped working and thus all users are now connected on it.

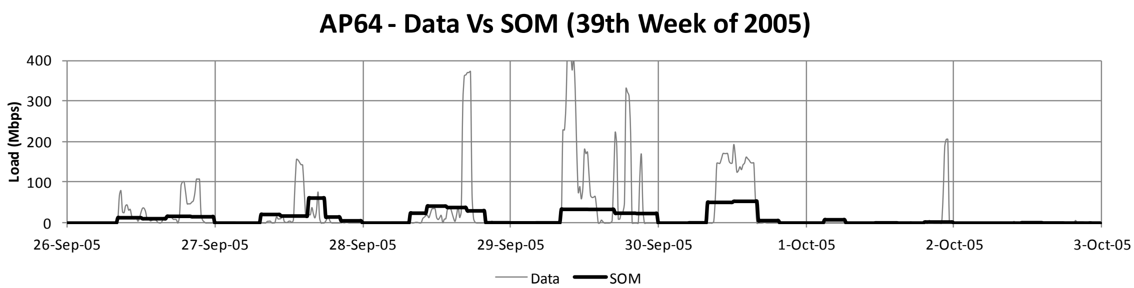
These are events that are not captured by any of the selected variables and thus the mechanism cannot learn them or predict them. A potential solution towards this direction, that will be considered in the future, is to offer the operator or the user of the mechanism the opportunity of informing the mechanism of such a change. Alternatively, a more autonomic solution would be to add a feedback loop to the mechanism that would inform it about the difference between the prediction and the actual load, as part of the online knowledge building. This would then insert a correction factor that would re-adjust the learnt pattern. This last solution would probably benefit a lot the results of the used dataset where the network load seems to increase a lot each year. Figure 5.14 captures the changes of the network load of one of the access point, namely of AP0, from year to year.











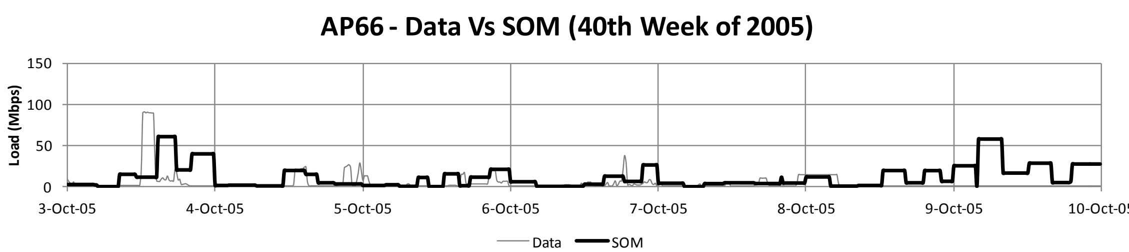
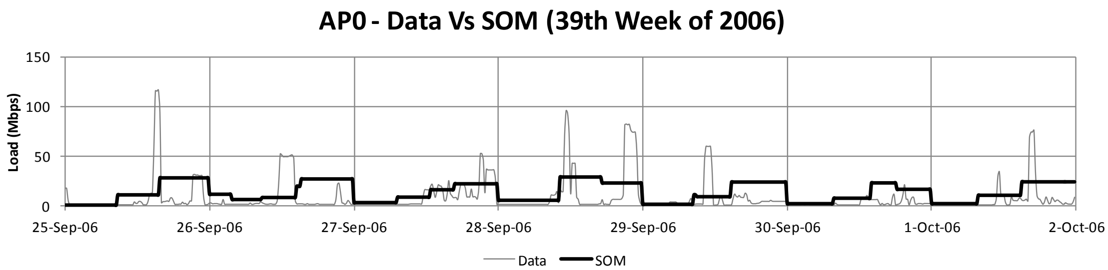
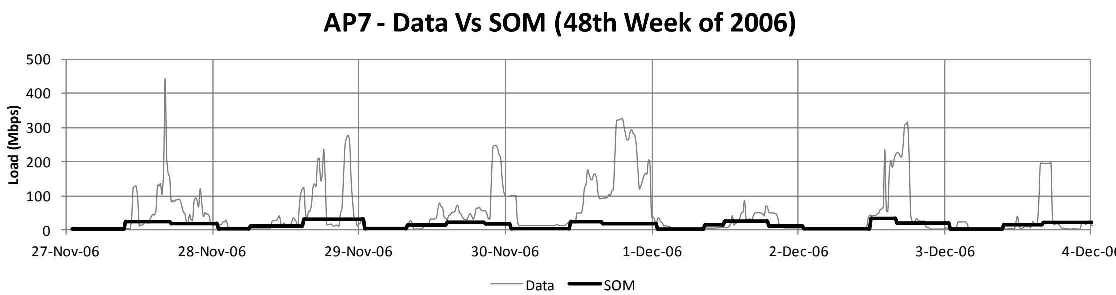
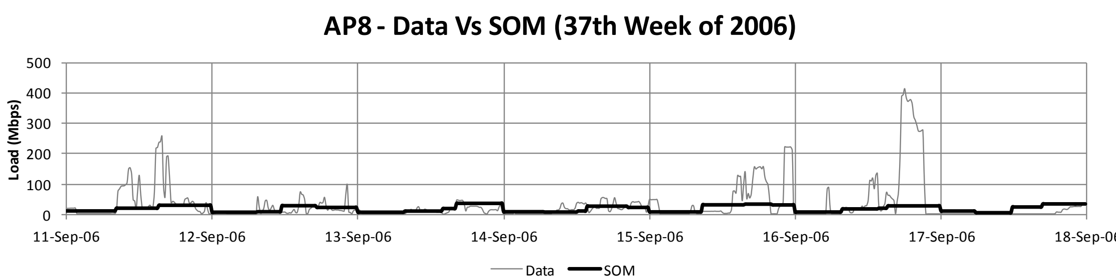
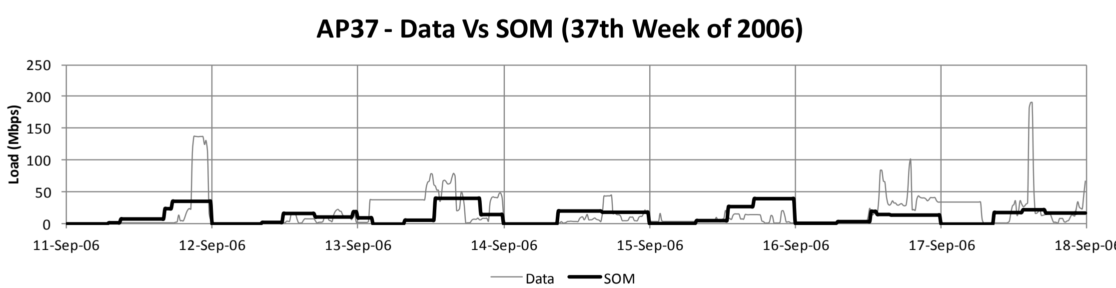


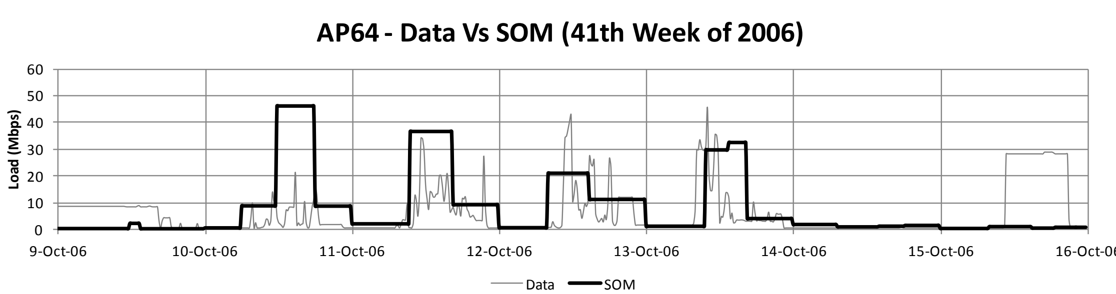
Figure 5.12: Indicative examples of comparative diagrams of real and predicted load values for each access point (AP0, AP7, AP8, AP37, AP64 and AP66) when using seen evaluation data.


AP0 - Data Vs SOM (39th Week of 2006)
Load (Mbps)
0
50
100
150
25-Sep-06
26-Sep-06
27-Sep-06
28-Sep-06
29-Sep-06
30-Sep-06
1-Oct-06
2-Oct-06
Data
SOM


AP7 - Data Vs SOM (48th Week of 2006)
Load (Mbps)
0
100
200
300
400
500
27-Nov-06
28-Nov-06
29-Nov-06
30-Nov-06
1-Dec-06
2-Dec-06
3-Dec-06
4-Dec-06
Data
SOM


AP8 - Data Vs SOM (37th Week of 2006)
Load (Mbps)
0
100
200
300
400
500
11-Sep-06
12-Sep-06
13-Sep-06
14-Sep-06
15-Sep-06
16-Sep-06
17-Sep-06
18-Sep-06
Data
SOM


AP37 - Data Vs SOM (37th Week of 2006)
Load (Mbps)
0
50
100
150
200
250
11-Sep-06
12-Sep-06
13-Sep-06
14-Sep-06
15-Sep-06
16-Sep-06
17-Sep-06
18-Sep-06
Data
SOM


AP64 - Data Vs SOM (41th Week of 2006)
Load (Mbps)
0
10
20
30
40
50
60
9-Oct-06
10-Oct-06
11-Oct-06
12-Oct-06
13-Oct-06
14-Oct-06
15-Oct-06
16-Oct-06
Data
SOM

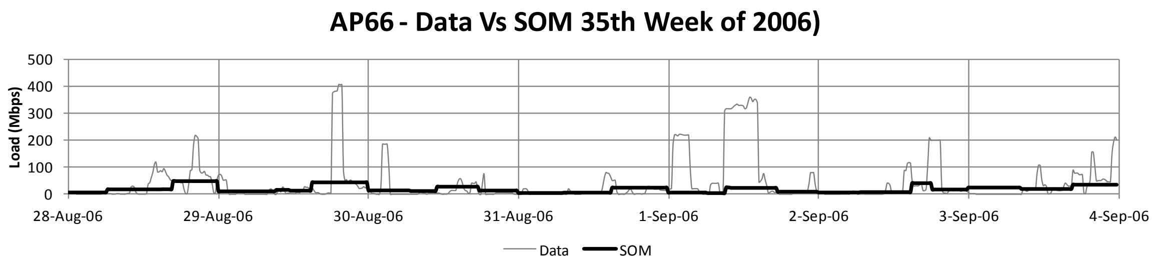
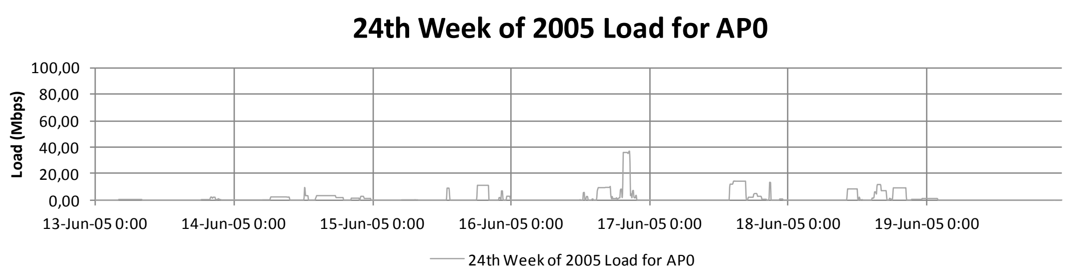
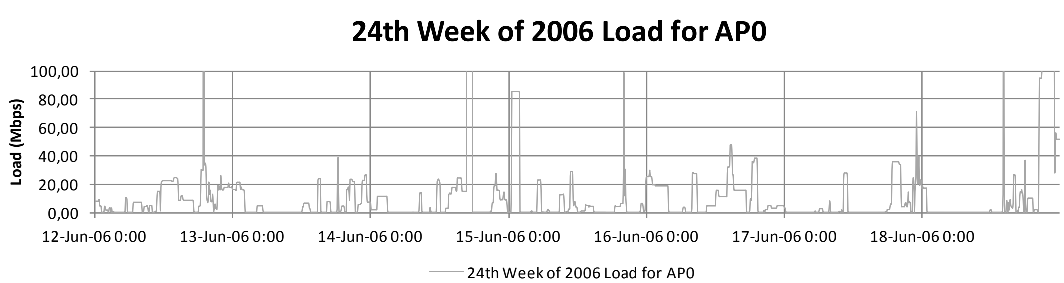


Figure 5.13: Indicative examples of comparative diagrams of real and predicted load values for each access point (AP0, AP7, AP8, AP37, AP64 and AP66) when using unseen evaluation data.





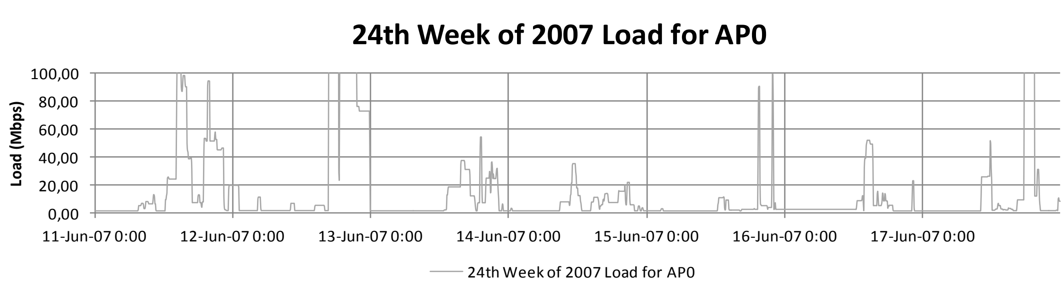


Figure 5.14: Diagram of the network load of AP 0 for the 24th week of the years 2005, 2006 and 2007.

5.3.4.2.3 Results with respect to the Big Data issues

The data that were exploited for this research were initially 3,574,080 unstructured numbers per year of the dataset. These data, during the pre-processing phase of the mechanism were organized in data samples, i.e., were transformed in 210,240 17D structured data per year of observations. When the information carried by these data samples was incorporated in the SOM of 900 cells, the only data that needed to be stored for maintaining all the information were the 900 vectors of the cells of the map.

Thus, the mechanism is capable of pre-processing and exploiting Big Data (in terms of disparity). Moreover, the dimensions of the data can be reduced from 17 to 2 during their visualization on the map. Last but not least, given the fact that the best performance was received by the map which was created using the data of 2 years and its size was equal to 900 cells, the mechanism offers the capability of reducing the size of the data that need to be stored from 420,480 to 900, i.e., approximately 467 times.

5.3.5 Scenario 3: Prediction of Network Traffic based on all the above parameters

This scenario exploits all 19 parameters presented in Table 5.2 (18 during the knowledge building) and respectively, the request towards the load prediction mechanism includes all of them but the load.

5.3.5.1 Data and Data Pre-processing

The data used for this scenario were the same as in the 2nd scenario (sub-section 5.3.4) while the same pre-processing applies as well. In this case, the data were complemented with weather and holidays. The information related to the mean temperature and the precipitation of each day from August 24th, 2004 up to August 24th, 2007 for the area of Montréal, Québec, Canada, i.e., the area where the access points are located, were retrieved from [9] using the Montréal-Pierre Elliott Trudeau International Airport as a reference point and a custom Java tool.

As soon as this was completed, for each load measurement there were data designating (i) the access point from which the load measurement had been monitored, (ii) the time,

(iii) the day, (iv) the week of the year, (v) the temperature, (vi) the precipitation during the day that the load was monitored and (vii) if that day was a holiday or not.

Eventually, the data had to become normalized and to be in a format recognizable by the SOM (Table 5.6), i.e., organized in data samples each of which contained one value for each of the 8 variables (access point ID, time, day, week of the year, temperature, precipitation, holidays and load measurement).

Table 5.6: Sample of data files that were inserted in the "Knowledge Building and Data minimization mechanism" in order to train the PLGSOM and build knowledge on the network load

Observation/ Data sample	AP0	AP7	AP8	AP37	AP64	AP66	Time	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Week of the Year	Mean Temperature	Precipitation	(bank) holiday	Load
1	1	0	0	0	0	0	48	0	0	1	0	0	0	0	5	20	0	0	54
2	0	0	0	0	1	0	300	0	0	0	0	1	0	0	9	15	1	0	10

To summarize, before performing the tests, a custom Java tool was used to pre-process the data that would:

- Read the trace file from [8], extract the information of the timestamp, the AP and the load, convert it in 1-7D variables and make the time interval of the observations equal to 15 minutes;
- Retrieve the temperature and the precipitation of the area from [9] so as to complement the observations;
- Retrieve from online calendars the (bank) holidays of the area;
- Normalize the values of the data; and
- Gather all the information in the form of data samples, i.e., in one file in which each row (observation) had information for the 8 variables, which had 1-7 dimensions each (see also Table 5.6).

This file was the one that was inserted in the "Knowledge Building and Data minimization mechanism" of sub-section 5.3.2.1 in order to train the PLGSOM and build knowledge on the network load.

5.3.5.2 Results

Both the validation and the metrics used in this scenario follow the approach of the previous 2 scenarios. Therefore, the results provides are divided in (a) qualitative results in the form of comparative diagrams between the predicted and the real values, (b) quantitative results using MSE, RMSE and MAE as metrics, and (c) results with respect to the efficiency of the tool to address some of the issues related to the Big Data management. Accordingly, for both the qualitative and the quantitative results two types of tests were performed a) those that used the same data for both building the knowledge and for the validation of the proposed tool, and b) those that used different data for the two functions/processes.

5.3.5.2.1 Qualitative Results

The qualitative results refer to comparative diagrams between the predicted and the real load values. Figure 5.15 depicts examples for each access point for (a) seen and (b) unseen evaluation data, respectively.

Similarly to scenario 2 (5.3.4), as it can be observed from the comparative diagrams of the figure in both types of tests, although the mechanism has learn the pattern of the load, i.e., when there is a load, it fails to predict some of its peaks. This may be caused by unpredictable events that suddenly attract (with no prior similar experience) more users, e.g., the place where the access point is hosted became more popular for some reason, e.g., openings of a new coffee shop next to it, or the access point next to it stopped working and thus all users are now connected on it.

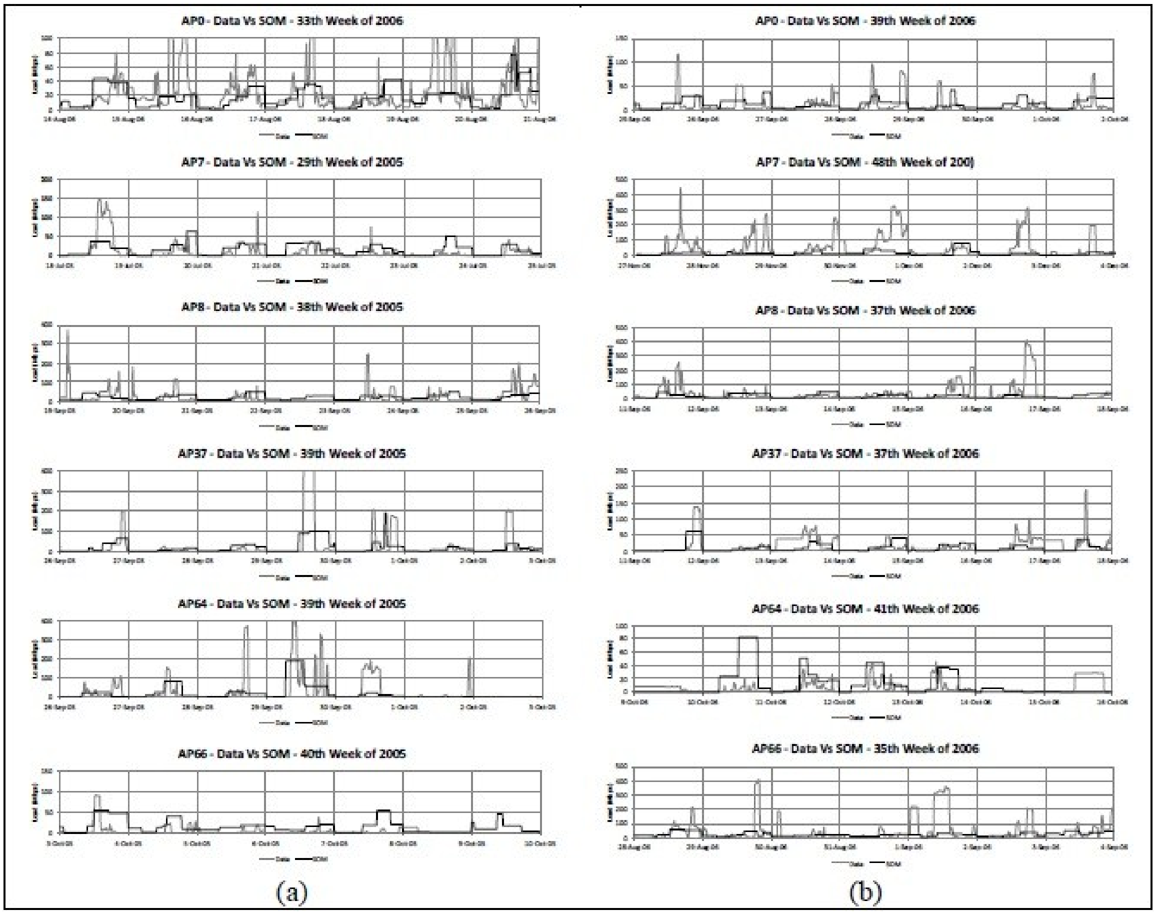


Figure 5.15: Indicative examples of comparative diagrams of real and predicted load values for each access point (AP0, AP7, AP8, AP37, AP64 and AP66) when using (a) seen and (b) unseen evaluation data.

These are events that are not captured by any of the selected variables and thus the mechanism cannot learn them or predict them. A potential solution towards this direction that will be considered in the future is to offer the operator or the user of the mechanism the opportunity of informing the mechanism of such a change. Alternatively, a more autonomic solution would be to add a feedback loop to the mechanism that would inform it about the difference between the prediction and the actual load, as part of online knowledge building. This would then insert a correction factor that would re-adjust the learned pattern. This last solution would probably benefit the results of the used dataset a lot where the network load seems to increase a lot each year. Figure

5.16 captures the changes of the network load of one of the access points, namely the AP0, over the years.

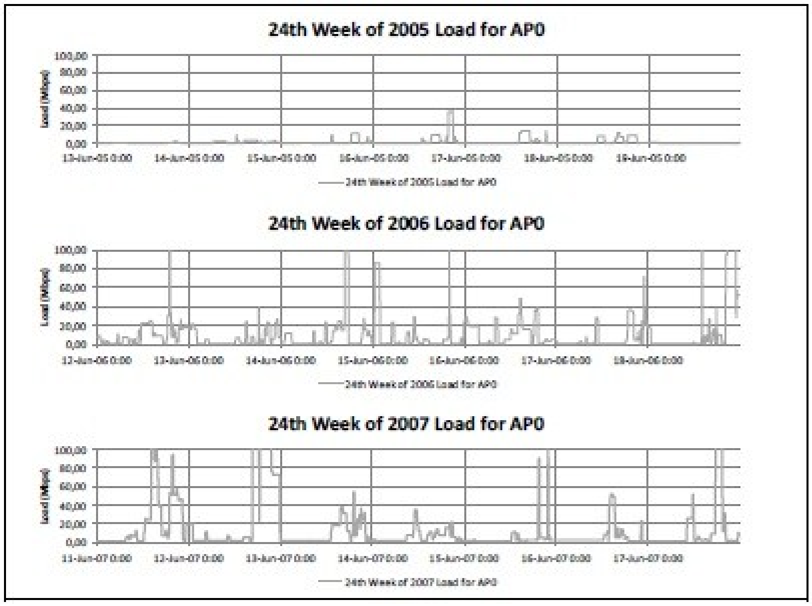


Figure 5.16: Diagram of the network load of AP0 for the 24th week of the years 2005, 2006 and 2007

5.3.5.2.2 Quantitative Results

Table 5.7 summarizes some indicative tests and their respective quantitative results. In the table, the data that were used for knowledge building are referred to as training data while those used for the evaluation of the mechanism are named evaluation data. Moreover, the table also points out the size of the map, measured in cells, which in the case of growing the SOM also represents how adjusted the map is to the training data.

Additionally, for the purposes of the evaluation of the mechanism, the initial dataset was split in three sub-datasets: (a) Y1 which refers to the dates from August 28th, 2004 up to August 28th, 2005, (b) Y2 involving data from August 28th, 2005 up to August 28th, 2006, and (c) Y3 which expands from August 28th, 2006 up to August 28th, 2007. For each of the created maps (1-3) two tests have been performed, one using the training data for the evaluation as well (1b, 2b, and 3b) and one having different training and evaluation data (1a, 2a, and 3a).

Table 5.7: Quantitative results

	Training Data	***Map Size***	***Evaluation Data***	***MSE***	***RMSE***	***MAE***
1a	Y1 & Y2	3465	Y3	0.000557	0.023607	0.007925
1b	Y1 & Y2	3465	Y1-Y2	0.000181	0.013469	0.005442
2a	Y2	2695	Y3	0.000578	0.024048	0.009601
2b	Y2	2695	Y2	0.000277	0.016655	0.007579
3a	Y1 & Y2	928	Y3	0.000553	0.023517	0.007925
3b	Y1 & Y2	928	Y1-Y2	0.000191	0.013817	0.005768

Comparing the results presented in the table, it must be noticed that the mechanism performs significantly better when the trigger refers to a case that has already been "seen" by the SOM. This observation was expected and that is also one more reason why online training has been considered.

Another useful observation is the fact that when only Y2 was used for building the knowledge (cases 2a and 2b); the performance of the mechanism deteriorates. This test was selected mainly for identifying if observations older than one year deteriorate the performance of the mechanism. According to the obtained results, it seems that older observations not only don't deteriorate the performance of the mechanism but they even enhance it. This is probably due to the fact that when using data only from Y2, less data have contributed to the knowledge building and thus, the SOM is familiar with fewer combinations of the involved variables.

The last observation, which comes from the comparison of 3a and 3b with 1a and 1b, respectively is that smaller maps result in better performance of the mechanism when the evaluation data differ from the training data and worse performance when the same data are used for both knowledge building and the evaluation. This is related to the fact that the bigger the map is, the more adjusted to the training data it becomes. In machine learning, this is also known as overtraining, which eventually results in the learning technique being unable to generalize well. For avoiding such cases, the performance of the technique between all the test cases (i.e., with seen or unseen data) should be kept as close as it gets. Thus, the best performance of the mechanism among the 3 presented in the table is considered to be the one using the 3rd map. Figure 5.17a depicts this map while Figure 5.17b-g depict the component maps of the access points and the days, the holidays, the time, the week of the year, the temperature and the precipitation, respectively.

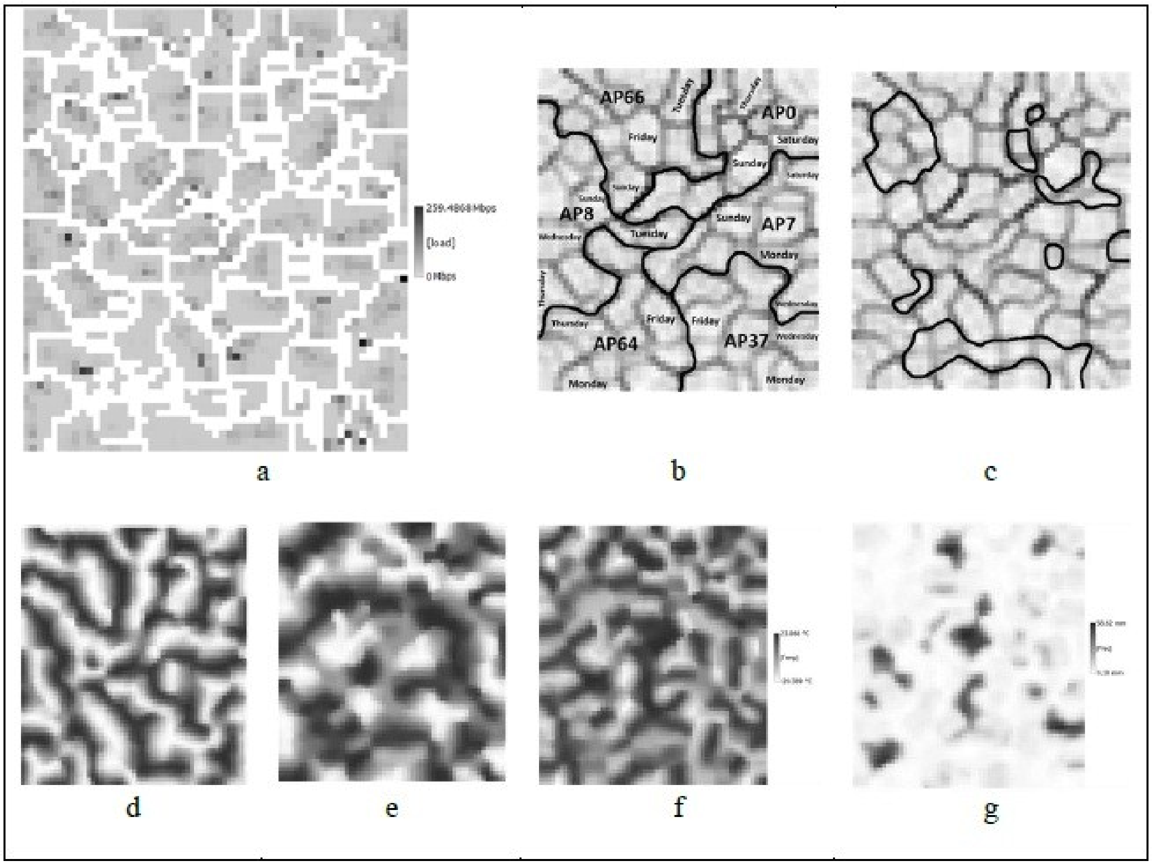


Figure 5.17: The map from which the best results were obtained from the (a) load, (b) access points and days, (c) holidays, (d) time, (e) week of the year, (f) temperature and (g) precipitation points of view. The marked areas in (c) refer to the holiday clusters while in (d), (e), (f), and (g) the higher the value of the respective parameter, the darker the area on which the data sample is mapped.

5.3.5.2.3 Results with respect to the Big Data issues

The data that were exploited for this research were initially 3,994,560 unstructured numbers per year of the dataset. These data, during the pre-processing phase of the mechanism were organized in data samples, i.e., were transformed in 3,994,560 / 19 = 210,240 structured data of 19 dimensions each per year of observations. When the information carried by these data samples was incorporated in the SOM of 928 cells, i.e., when the data were mapped on the SOM, the only data that needed to be stored for maintaining all the information were the 928 vectors of the cells of the map.

Thus, the mechanism is capable of pre-processing and exploiting Big Data (in terms of disparity). Moreover, the dimensions of the data can be reduced from 19 to 2 during their visualization on the map. Last but not least, given the fact that the best performance was received by the map that was created using the data of 2 years and its size was equal to 928 cells, the mechanism offers the capability of reducing the size of the data that need to be stored from 2*210,240 = 420,480 to 928, i.e., approximately 453 times.

5.3.6 Comparison of the considered scenarios

Table 5.8 summarizes the best obtained results from the 3 scenarios in the case of unseen data in terms of denormalized MAE.

Table 5.8: Summary of the results obtained from the three scenarios

	Denormalized MAE
Scenario 1	1.198 Mbps
Scenario 2	25.652 Mbps
Scenario 3	26.030 Mbps

According to the table, the best results were obtained when apart from the main parameters, i.e., the area, the time, the day, and the load, only weather was taken into account. However, as the datasets used in the first scenario (main parameters + weather conditions) and the last 2 scenarios were different, the outcome needs to be re-validated with the same dataset for all three scenarios. Moreover, user preferences may not be equally influenced in all geographical areas by all these factors, e.g., in northern cities/countries, the users' intention to use the network may be less influenced by an upcoming (bank) holiday than in southern cities/countries.

5.4 Conclusions

One challenge in the area of cognitive management is the design of mechanisms that can derive meaningful, as well as useful information for the stakeholders or other intermediate systems and mechanisms in an efficient and automated fashion. Often, another challenge is the storage of huge amount of disparate data (Big Data) which, depending on their volume, might be prohibitive. In particular, in telecommunication networks, huge amounts of data can be collected when, for instance, traffic-related parameters are measured on a high granularity, for long time periods and for multiple nodes. The NO is then forced to maintain measurements in a certain time-window and discard the ones that fall out of it. As a result, network operators may collect large amounts of historical data related to the load of the network that need to get organized in less data while maintaining or even augmenting/enhancing the quality of the information. In this case, the effective removal of redundancy (i.e., compression) either lossy or lossless depending on the problem and the transformation of the data in higher level information, so as to become more easily usable by the NO, is the solution to the challenge.

To this end, this chapter presented two mechanisms that analyse network and human-oriented data so as to build and provide knowledge with respect to the either the possibility of a link to get congested or the traffic of a node. The data used for building knowledge in the first mechanism were exclusively directly monitored by the network, i.e., refer to network parameters, and thus allow narrow timeslots for proactively overcoming a congested link. To address this issue, the training data of the second mechanism takes into account human-oriented parameters (e.g., time, date, location, etc.) as well - allowing to also foresee more long-term situations and adjust the network parameters accordingly.

5.5 Chapter References

[1] Kuremoto, T., Komoto, T., Kobayashi, K., Obayashi, M.: Parameterless-Growing-SOM and Its Application to a Voice Instruction Learning System. J. Robotics (2010). doi:10.1155/2010/307293

[2] CRAWDAD database, available at http://crawdad.cs.dartmouth.edu/index.php, last accessed: July 23rd, 2012

[3] Dataset: dartmouth/campus (v. 2009-09-09), available at http://crawdad.cs.dartmouth.edu/meta.php?name=dartmouth/campus, last accessed: July 23rd, 2012

[4] Trace: dartmouth/campus/snmp/fall0304 (v. 2004-11-09), available at http://crawdad.cs.dartmouth.edu/meta.php?name=dartmouth/campus#N1019E, last accessed: July 23rd, 2012

[5] The Old Farmer's Almanac website, weather history, http://www.almanac.com/weather/history, last accessed: July 23rd, 2012

[6] CRAWDAD database: http://crawdad.cs.dartmouth.edu/index.php. Accessed 23 July 2012

[7] The Île Sans Fil website: http://www.ilesansfil.org/. Accessed 10 August 2012

[8] CRAWDAD database: Trace ilesansfil/wifidog/session/04_07 (v. 2007-08-27) http://crawdad.cs.dartmouth.edu/meta.php?name=ilesansfil/wifidog#N1006B. Accessed 10 August 2012

[9] Wunderground.com Website: http://www.wunderground.com/history/. Accessed 15 August 2012

6 COGNITIVE DATA ANALYTICS IN TRANSPORT LAYERS

6.1 Chapter Outline

Today's large development of networks increases their complexity in levels which sometimes become difficult to manage. An often proposed way for addressing this issue is to enable networks with cognitive/ autonomic capabilities usually achieved by Self-x mechanisms. Although these mechanisms can receive autonomic decisions given the past experience of the system and its current state they are not in position to ensure stability of the system. On the other hand, network instabilities should be avoided as they may lead to both worse network performance and higher utilization of resources. This chapter studies the problem of network stability by defining it and identifying some of the basic reasons that may lead to it. Moreover, focusing on instabilities that come from congestion control mechanisms, a learning based method is proposed for enhancing their decisions and thus increasing network stability. In particular, SOM is used as basis for building knowledge on the performance of the congestion control mechanism TCP Vegas. This knowledge is then used for "consulting" the congestion control mechanism when adjusting for enhancing its functionality and avoiding congestion in a more stable way. The first tentative results suggest that this kind of knowledge may indeed enhance the performance of TCP Vegas by eliminating some of its drawbacks and by assisting the latter to work in a way that allows higher and more stable network utilization. The keywords of this chapter are: stability, machine learning, congestion control, TCP Vegas, Self-Organizing Maps (SOMs)

6.2 Introduction

Today's large exploitation of networks in almost all facets of everyday life increases the complexity of their management. Towards this direction, self-x mechanisms that are incorporated in cognitive/ autonomic networks can offer relaxation. However, autonomic decisions cannot always ensure stability in networks. Instabilities may occur due to three main reasons. The first reason refers to the local decisions (made per node), which do not always take into account decisions of neighbouring nodes, thus leading often to a globally unstable situation. Moreover, contradictory technical and/ or business objectives can also result in the same undesired situation. The last but not least important reason comes from the trade off of different goals. For example, a goal that targets at congestion avoidance may contradict to a goal for stability in the network utilization and best resource exploitation.

As such situations, i.e. instabilities of the network, can both jeopardize the performance of the network and compromise an optimized use of resources, studying, understanding and avoiding them becomes necessary for designing and exploiting the autonomic/ cognitive networks. The latter will enable networks to make decisions with respect to the stability levels that can be achieved in both smooth and transient phases.

This study focuses on instabilities that derive from congestion control mechanisms when applied in a dynamic network and introduces a methodology for observing network behaviour and building knowledge with respect to it. In particular, the chapter studies the instabilities of the network utilization and the resource exploitation that come from the TCP Vegas algorithm [1] when trying to avoid congestion proactively (minimizing the dropped packets) and not reactively, as most congestion control mechanisms do. Studying this drawback of TCP Vegas, enabled us to better understand what kind of knowledge is missing, built it with the unsupervised learning technique of SOM (sub-section 3.4) and enhance the performance of TCP Vegas in terms of selected congestion window and network utilization. As a result, applying TCP Vegas maintains the reactivity of the mechanism without causing that much instability to the congestion window, the rate of the TCP flow and thus the utilization of the network.

The following sub-sections present an overview of two different existing congestion control mechanisms and highlights how these can lead to instabilities of the network. Similar issues appear in other congestion control mechanisms, although they are not explicitly presented here for brevity reasons. In the sequence, the problem statement, the approach and the respective results are introduced.

6.3 Related Work

Congestion control is an example of how local rules that control the way that the packets are transmitted through a network can correspond with maximizing of aggregate utility across the entire network. Actually, congestion control mechanisms represent one of the largest deployed artificial feedback systems: a global network feature is the consequence of local rules.

Currently available congestion control mechanisms (like TCP Reno [4], New Reno [5], Cubic [6], Tahoe [7] and Vegas [1]) are examples of large distributed control loops deriving the state for a desired equilibrium point but they do not take into account transient behaviours (which, on the other hand, are typical in closed-loop systems). For example, Reno congestion avoidance is a sort of reactive scheme that determines available network bandwidth by packet loss. Reno increases the congestion window linearly during congestion avoidance and reduces the congestion window when either three duplicate ACKs are received or a coarse grained timeout occurs. Reno makes no attempt to detect, predict network congestion, and as a result, the goodput (i.e., the ratio of bytes transmitted excluding duplicates to total bytes transmitted) of connections is not optimized. Similar approaches are followed by New Reno, Cubic and Tahoe as well.

As another example, TCP Vegas detects network congestion in the early stage and successfully prevents periodic packet loss that usually occurs in the above mentioned congestion control mechanisms. TCP Vegas performs better than those in terms of packet losses but it suffers several problems such as those analysed in sub-section 6.3.

Overall, these mechanisms don't take account of transient behaviours. Thus, these mechanisms may be ill-suited for future dynamic network where delays and capacity can be large. Even worst, this can result in potential instabilities in networks, which can have primary effects both in jeopardizing the performance and compromising an optimized use of resources.

Let's consider the problem in more details. Overall a network can be modelled as the interconnection of resources/links carrying the data generated by users/sources. Associated with each source is a route, which is the collection of links through which information from that source is flowing. In traditional congestion control, each link sets a price per unit of flow, based on the aggregate flow crossing that link, and the sources set their transmission rates based on the aggregate price they detect. In the absence of delays, this scheme is globally stable. In the presence of delays (or large bandwidth) the scheme can become unstable.

Sources try maximizing individual profit based on their own utility functions. On the other hand, links use prices to align, exactly or approximately, sources "selfish" behaviour.

One way to approach stability (for this example) is to define the Lagrangian function, so that the global optimization problem (flow control) can be turned into the search of saddle point problem. State stability, in this case, can be defined by a Lyapunov function. On the other hand, it should be noted that an inadequate Lyapunov function may cause the excess of false-positive warnings, a risk that cannot be avoided. Moreover even if we obtain an asymptotic stability, it is not clear what may happen during transients (typical of control feedbacks).

This problem is even more complicated when it is desired to accomplish both routing and congestion control simultaneously. The objective in this case is to find an optimal route for every source so that the utility function of the network is maximized. Since future networks are expected to be highly dynamic (with nodes joining and leaving the network with different time scales) it is important to take account of the transient network behaviours when optimizing routing and congestion strategies.

Mechanisms and methods for optimizing congestion control performance and stability even during transients are missing; this self-adaptive behaviour should be based on measurement-estimation-prediction of network conditions (e.g. by using RTTs, or other network indicators). The proposed way forward in this study involves a learning mechanism capable of building knowledge on the network behaviour with respect to the decisions made by the congestion control mechanism. This knowledge is fed back to the congestion control mechanism so as the latter to become aware of the results of its past decisions and to enhance the future ones by predicting their results according to this past experience. The exact problem to be studied is further explained below while some first tentative results follow.

6.4 Problem Statement

As presented above for the two examples of congestion control mechanisms, their drawbacks can lead to instabilities of the network. Motivated by the need of avoiding unstable behaviours of the system, a methodology for observing and building knowledge on the network behaviour, when applying a congestion control mechanism, is proposed hereafter. In particular, the studied congestion control mechanism is the TCP Vegas.

This congestion control mechanism, although it is capable of acting proactively for minimizing the dropped packets, it has a very important drawback that has led to it not being widely applied. This disadvantage has its roots to the inefficiency of the algorithm to distinguish the reason that caused an increase of the measured RTT. Thus, every such increase is perceived as a congested link leading to the decrease of the congestion window so as to avoid the congestion. However, other reasons may have caused this increase. Such reasons are possible reroutes in the forward [4] or the backward [5] path of the TCP flow.

This misinterpretation makes TCP Vegas decrease its congestion window instead of maintaining its size or even increasing it. On the other hand, if TCP Vegas had the information that this was not a congested link but another reason, then its minimum RTT would have been updated with the increased value instead of being kept the same. What is attempted here, is to build knowledge of when TCP Vegas should be reset and when not to so as it always has the correct minimum RTT.

This knowledge can then be used as a feedback to the congestion control mechanism, urging the mechanism to reset and thus update its minimum RTT when needed. Consequently, this will eventually result in the highest possible rate of the TCP flow and thus network utilization with simultaneous avoidance, as much as possible, of the dropped packets and to a more stable way of acting for the TCP Vegas.

The overview of the proposed methodology is summarized in Figure 6.1 while a more detailed description follows in the next sub-section.

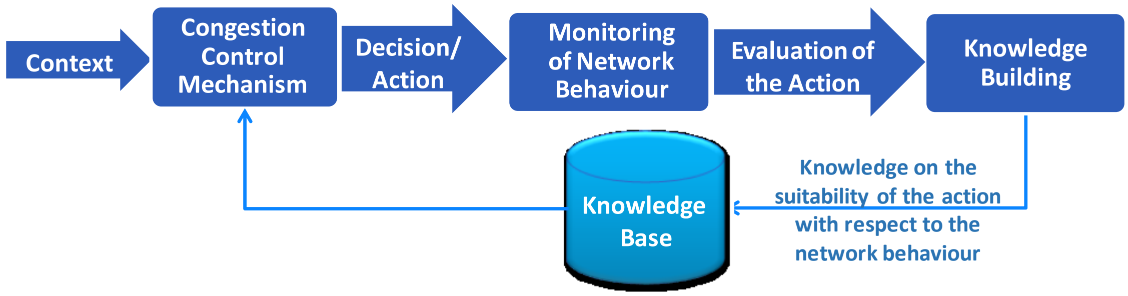


Figure 6.1: Methodology for building knowledge on when the TCP Vegas needs reset/ update of its minimum RTT.

6.5 Approach

The adopted approach is depicted in Figure 6.2 and is as follows: the congestion control mechanism monitors the network changes and queries the knowledge base (the map created by the SOM technique in our case) whether it should be reset or not each time the RTT changes more than a threshold.

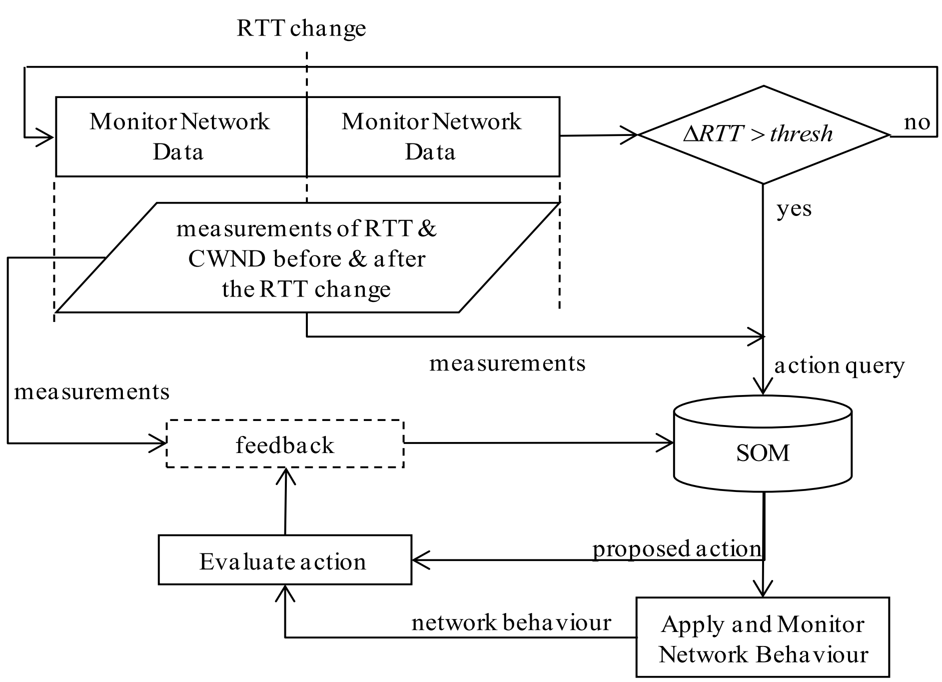


Figure 6.2: The followed approach for learning whether a reset is needed or not.

In order to built this knowledge, SOM has performed vector quantization [2] so as to cluster input data and generalize on the patterns they follow regarding the two actions i.e., reset/maintain minimum RTT. Thus, by mapping new measurements (i.e., RTT and congestion window measurements before and after the RTT change) to the created SOM, the algorithm can provide an action proposal to the TCP mechanism with respect to its past experiences. TCP Vegas applies the action suggested by SOM and keeps monitoring. The monitored data after the taken action comprise the network behaviour. Network behaviour, the monitored data before the SOM interval and the proposal of the knowledge base are then fed back to SOM comprising its online training. It should be clarified here that initially when the monitored data describe a context that SOM has not seen again in the past, SOM proposal to TCP Vegas is to reset, i.e. SOM default proposal is to reset. As time passes, due to the feedback that SOM receives and its online training, the past experience and the built knowledge increases. Thus, SOM becomes aware of more different situations and capable of giving more accurate and reliable proposals.

6.6 Results

In order to examine the performance of the proposed method, simulations using Network Simulator 2.35 were performed. The network topology simulated is depicted in Figure 6.3. In this topology, nodes 3, 4 and 5 are routers and nodes 1, 2, 6 and 7 are end-hosts. In particular, node 1 is an FTP server, node 6 is an FTP client connected to the server and continuously downloading from it, while node 2 runs a UDP source application sending data to node 7 at specific time period.

Although the network topology is simplified, it is adequate enough in reproducing and investigating our problem, since the number of hops in a path is not directly affecting Vegas' underestimations of RTT. Our key factor here is the total RTT measured by the end-to-end hosts, and thus, there is no loss of generality even in this simplified topology.

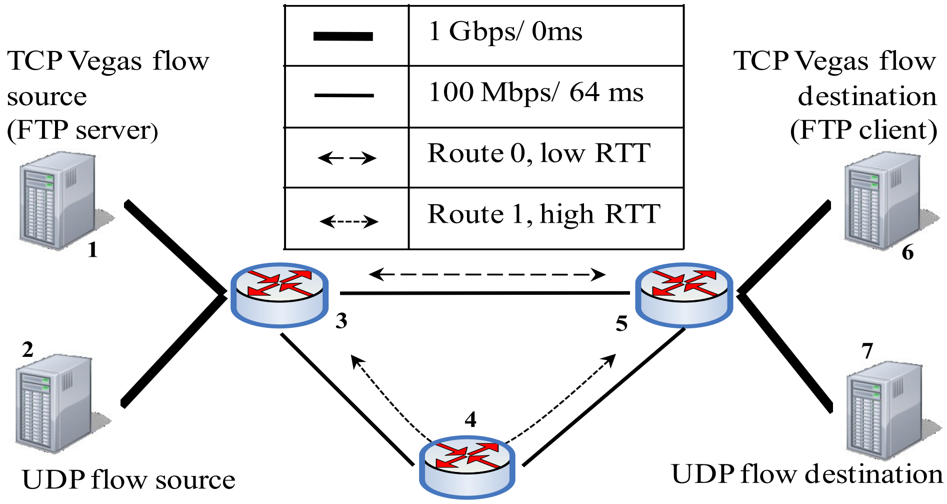


Figure 6.3: Network configuration for the simulations.

Six test cases were examined and organized in two simulation scenarios. The first simulation scenario targets at showcasing three basic test cases in which TCP Vegas misinterprets the monitored increase of the RTT for congestion. This results in TCP Vegas decreasing its congestion window instead of maintaining or increasing its size. In particular, the default route of the scenario for both the forward and backward path for the TCP-FTP flow between the nodes 1 and 6 is route 0. The TCP flow starts simultaneously (t_0=0 sec) with the simulation. Enough time is given to the TCP sender so as to increase its congestion window to its maximum and stabilize. At time t_1=200 sec the forward path of the TCP-FTP flow is rerouted through node 4 (route 1), which has higher minimum RTT than route 0 (first test case). However, Vegas has no indication of this change and since the minimum RTT has higher value than the previous one, Vegas fails to identify this fact and update it. Thus, the only information that Vegas has is that the RTTs have increased, which for Vegas designates a congested link. So the congestion window is decreased, instead of the opposite, so as to avoid the congestion. At t_2=450 sec the path is reverted back to route 0. At time t_3=650 sec, a UDP flow starts between end points 2 and 7. The sending rate (from node 2 to node 7) is initially set to 1 Mbps and is then increased by 0.5 Mbps per second for a time period of 200 seconds. In this case as well, Vegas detects an increase to the RTTs as a signal that there is a traffic increase and decreases the congestion window. Although this decision is correct, this will eventually end up to UDP flow dominating the link between nodes 3 and 5. At time t_4=850 sec, link 3-5 becomes congested and the TCP-FTP flow is rerouted through node 4 (route 1) (second test case) while at the same time the UDP flow stops. In this case as well, just like the first test case, Vegas failed to understand that the route has changed and that the minimum RTT needs to be updated ending up in no reaction regarding the congestion window size (grey line of Figure 6.4). At t_5=1100 sec, the TCP-FTP flow gets back to route 0, TCP Vegas realizes the decrease of the RTTs and starts increasing again the congestion window. On the contrary, at t_6=1350 sec, when the backward path is rerouted through node 4 (third test case), Vegas does not identify the cause of this change. Once more, it misinterprets its observation for congestion and thus decreases again the congestion window that in this case should have increased. The new reroute of the forward path of the TCP-FTP flow through node 4 at t_7=1550 sec, as expected from our previous observation, results in no increase of the congestion window. Contrarily, it has reached its lowest value equal to 0. Finally, the simulation ends at t_8=1850 sec. The corresponding fluctuation of the observed congestion window in the above scenario is also depicted in Figure 6.4.

In the same figure (Figure 6.4) is also depicted the same scenario supported by knowledge. As explained above, the knowledge is built following a methodology based on SOM learning technique. This knowledge enhances the decisions of Vegas when a large RTT is observed. In particular, SOM encourages or discourages Vegas to reset according to the past experience of the network and its current context. As can be seen in Figure 6.4 knowledge has intervened in Vegas decision at t_1=207 sec, t_2=857 sec, t_3=1357 sec and t_4=1558 sec, i.e., when the increase of the RTTs were not caused by congestion but by other circumstances. The comparison of the two diagrams in Figure 6.4 reveals that knowledge can indeed eliminate the faulty assumptions of Vegas improving its performance in terms of increasing and maintaining the size of the congestion window instead of decreasing it.

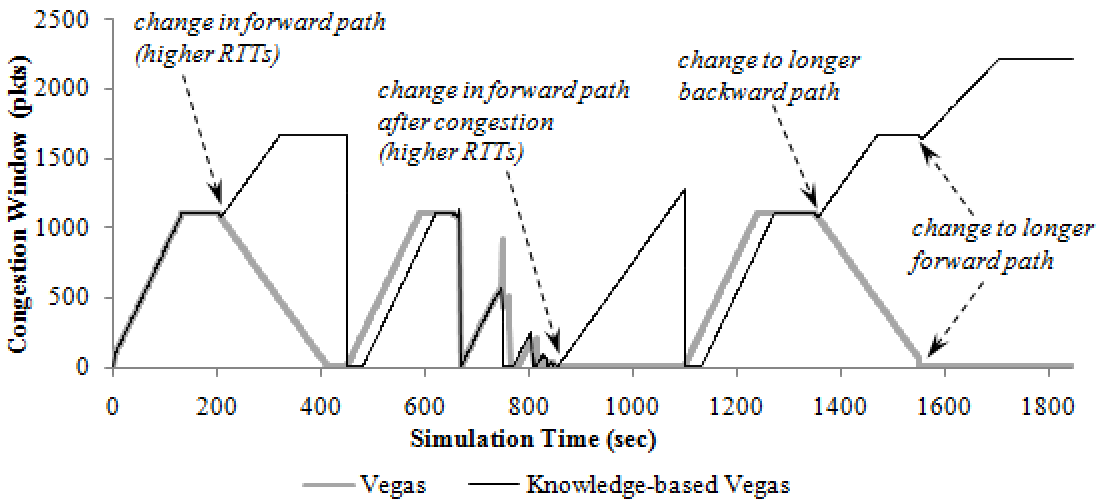


Figure 6.4: Fluctuation of the congestion window when using TCP Vegas with and without the support of SOM (Scenario 1).

Accordingly to congestion window size, Figure 6.5 depicts the fluctuation of the rate of the TCP flow both with and without support. As the capacity of the path is restrained by the links with the minimum capacity regardless the selected route, the maximum rate that can be achieved is 100 Mbps. Moreover, as can be seen in the diagrams, faulty assumptions of TCP Vegas result in not exploiting in all cases the maximum capacity of the link. On the contrary, when knowledge is used in support of Vegas, maximum capacity of the link is used for more often and for longer periods of times. Thus, support of Vegas by knowledge provides a higher and more stable rate for the TCP flow.

Moving to the second simulation scenario, the following changes were done with respect to the first one. The acknowledgements are sent following the delayed acknowledgement technique with a delay of 200 ms. Moreover, the propagation delay of the links between the nodes 3, 4 and 5 was increased to 128 ms while the capacity of the same links was decreased to 10 Mbps. Accordingly, the UDP flow rate was initially set to 0.2 Mbps while the rate with which it was increasing was 0.1 Mbps per 100 sec. The behaviour of Vegas for this scenario in terms of congestion window size and achieved rate of the TCP flow are depicted in Figure 6.6 and Figure 6.7, respectively. Similarly to the conclusions in which we were led by the 1st scenario, this scenario showcased that the performance of TCP Vegas is significantly improved both in terms of congestion window size and achieved rate of the TCP flow when the mechanism is supported by knowledge.

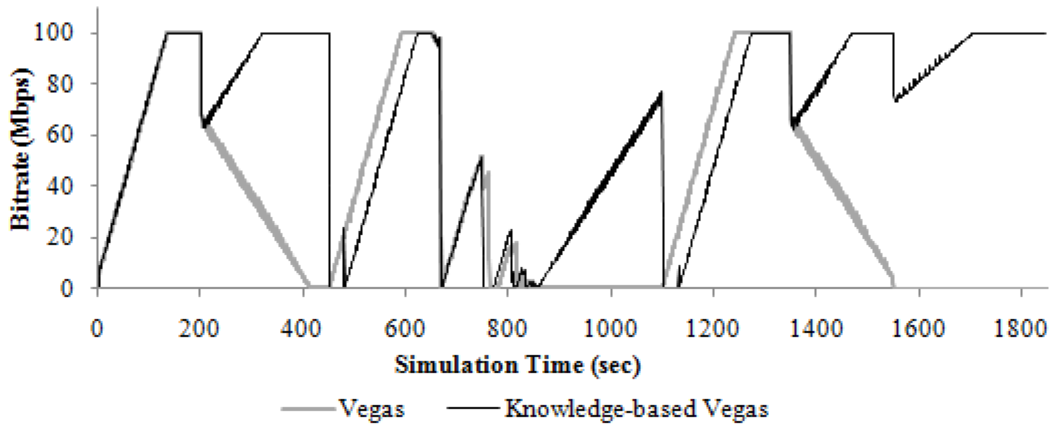


Figure 6.5: Fluctuation of the bit-rate of the TCP flow when using TCP Vegas with and without the support of SOM (Scenario 1).

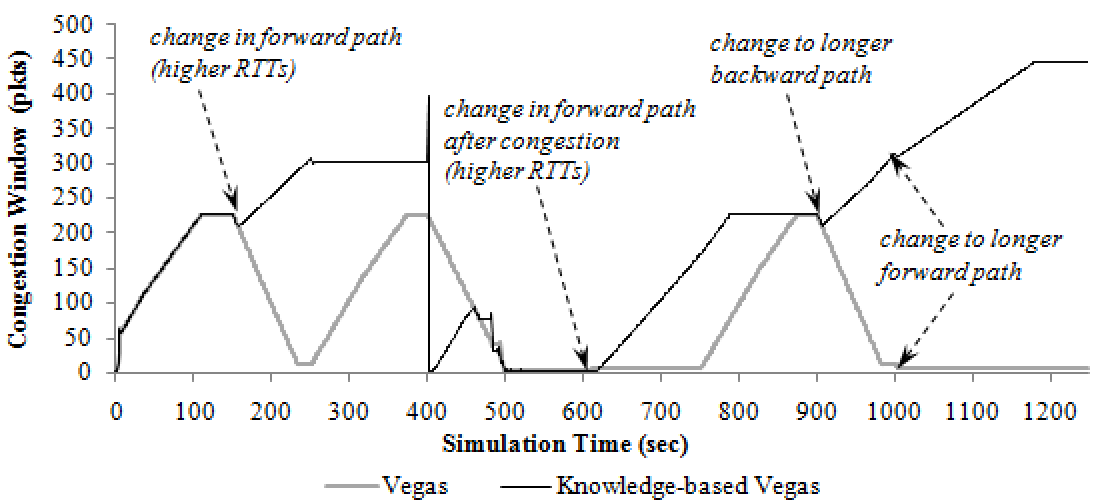


Figure 6.6: Fluctuation of the congestion window when using TCP Vegas with and without the support of SOM (Scenario 2).

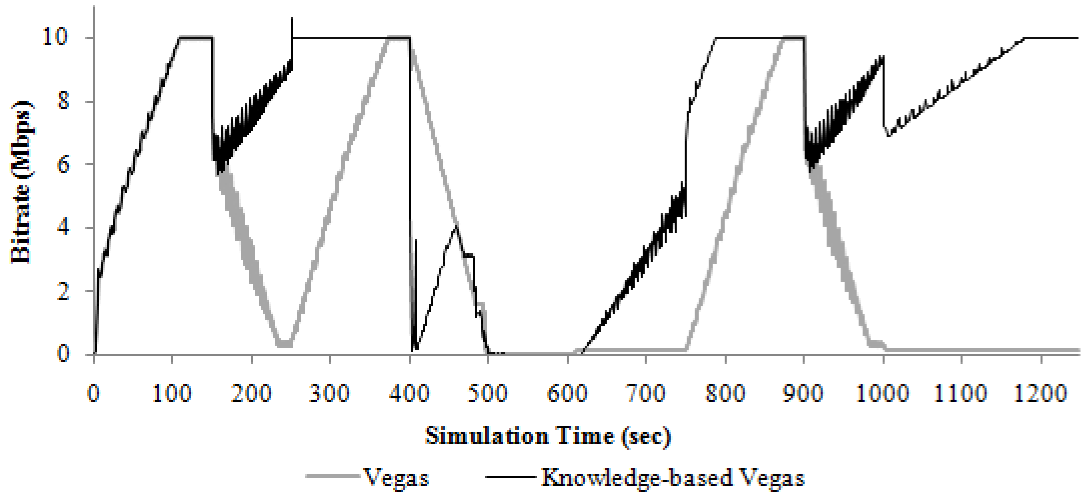


Figure 6.7: Fluctuation of the bit-rate of the TCP flow when using TCP Vegas with and without the support of SOM (Scenario 2).

6.7 Conclusions

The pervasiveness of networks in the everyday life of humans makes network complexity increase. This phenomenon leads to the need of applying more autonomic/ cognitive solutions for the management of the networks. On the other hand, autonomic/ cognitive management solutions cannot ensure the stability of the network, especially in dynamic network environments. Additionally, instability issues that may accrue in such cases jeopardize the network performance and the maximization of resource utilization.

As a result, stability issues need to be identified and studied, even in transient phases of the network so as instabilities to be avoided. This study targeted at stability issues that origin from congestion avoidance mechanisms, and in particular from TCP Vegas, and offered a proposal for enhancing the latter through learning. Learning techniques, such as SOM that was used in this study, are capable of building knowledge with respect to actions that need to be taken by the congestion control mechanisms given the situation encountered in the network.

The approach followed here was to provide the built knowledge to the TCP Vegas so as to enhance the rate of the TCP flow maintaining TCP Vegas attribute of proactively (in terms of dropped packets) avoiding congestion. The first tentative results proved to be encouraging towards the proposed methodology. However, more research needs to be done so as to achieve better results and even more stable actions. In particular, as this research is ongoing, the future work involves the study of more scenarios and even the enhancement of the methodology towards the desired direction.

6.8 Chapter References

[1] Brakmo, L.S.; Peterson, L.L.;, "TCP Vegas: end to end congestion avoidance on a global Internet," Selected Areas in Communications, IEEE Journal on , vol.13, no.8, pp.1465-1480, Oct 1995. doi: 10.1109/49.464716

[2] Kohonen, T.; , "The self-organizing map," Proceedings of the IEEE , vol.78, no.9, pp.1464-1480, Sep 1990. doi: 10.1109/5.58325

[3] Berglund, E.; Sitte, J.; , "The parameterless self-organizing map algorithm," Neural Networks, IEEE Transactions on , vol.17, no.2, pp. 305- 316, March 2006. doi: 10.1109/TNN.2006.871720

[4] K. Fall and S. Floyd, "Simulation-based comparisons of Tahoe, Reno, and Sack TCP", ACM Comput. Commun. Rev., Vol. 26, No. 3, pp.5 - 21 , 1996

[5] S. Floyd, T. Henderson, and A. Gurtov, "The NewReno Modification to TCP's Fast Recovery Algorithm", RFC3782, April 2004

[6] Sangtae Ha, Injong Rhee, Lisong Xu, "CUBIC: a new TCP-friendly high-speed TCP variant", SIGOPS Oper. Syst. Rev., Vol. 42, No. 5. (July 2008), pp. 64-74. doi:10.1145/1400097.1400105

[7] Jacobson, V.; , "Congestion avoidance and control," ACM SIGCOMM Computer Communication Review , vol.18, no.4, pp.314-329, August 1988 doi: 10.1145/52325.52356

[8] Mo, J.; La, R.J.; Anantharam, V.; Walrand, J.; , "Analysis and comparison of TCP Reno and Vegas," INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE , vol.3, no., pp.1556-1563 vol.3, 21-25 Mar 1999. doi: 10.1109/INFCOM.1999.752178

[9] Yi-Cheng Chan; Chia-Tai Chan; Yaw-Chung Chen; , "An enhanced congestion avoidance mechanism for TCP Vegas," Communications Letters, IEEE , vol.7, no.7, pp. 343- 345, July 2003. doi: 10.1109/LCOMM.2003.814715

7 TRUST DEVELOPMENT AND MANAGEMENT MECHANISMS

7.1 Chapter Outline

As presented in the chapters above, learning capabilities are somehow interwoven with cognitive systems. In particular, they are expected to incorporate past experience and knowledge of the network in the system and thus facilitate their decisions. Moreover, they are expected, and in some cases have proved their ability, to enable faster decisions which are not any more "blind", in terms of not knowing the expected results. In these terms, learning capabilities will enhance the automation of network decisions with respect to their past and the time needed for reaching them.

Additionally, learning capabilities are expected to bring forward benefits to both NOs and end users. More specifically, and regarding NOs, building knowledge will reduce both CAPEX and OPEX. CAPEX is the NO's expenditures that are related to technological investments or equipment so as to enhance or maintain the ability of offering the services and serving the demand. For example, building knowledge on either user preferences/ behaviour regarding the services and his demands or the network capabilities could benefit decisions with respect to expanding or not the network equipment. Moreover, decisions related to the distribution of the traffic can also be improved by taking into account such knowledge. In these terms, correct network and service planning will eventually decrease CAPEX.

On the other hand, OPEX refers to NO's expenditure coming from human resources and the operations of the equipment in general. Towards this direction, OPEX shall be decreased in two ways. The first one derives from decreasing human resources. Applying learning mechanisms will make automatic decision making mechanisms more applicable and reliable when using past experience of the network, thus less human resources will be needed and hopefully less human mistakes will occur. The second benefit regarding OPEX comes from the configuration of the equipment so as to function using the minimum required energy. For example, suitable network planning, contrarily to the current worst case scenario planning, could be applied with respect to a green footprint, i.e. reducing needed energy. The latter, apart from the social economic aspect, will eventually also reduce OPEX.

Moving towards users end, learning is expected to benefit them as well. In particular, decisions enabled by some of the examples that have already been mentioned are also expected to increase offered QoS and QoE. The first aspect is rather obvious as it is already associated with the proper distribution of the services and the traffic and capabilities of the network. For example, selection of the most appropriate network configuration, with respect to the data rate that it can offer, results in better QoS towards users. On the other hand, learning capabilities in a network can also enhance QoE. Imagine only a network that would have the ability to predict future faults, in the near or the distant future, and resolve such issues before even user experiences them. Such a feature would definitely improve users' QoE.

Although benefits seem to be many, moving from human handled networks to cognitive ones needs cautious and stable steps. Despite the fact that learning is capable of enhancing network decisions, applying them can turn against the network in terms of complexity. Thus, caution is needed when choosing the learning technique that will develop each type of knowledge, and the respective variables that will reveal the context where the network operates. Moreover, a challenging issue arises when considering cross-layer and cross-domain configurations, as omitted variables may provoke non-linear behaviour of the latter and instabilities when training the system. Finally, although autonomicity facilitates decision making in CRS and enhances them in terms of speed, caution is also needed for ensuring that the decision made indeed result in better performance of network. Trust to the autonomic and self-x mechanisms need to be built. To this end, this chapter proposes a knowledge building mechanism that is expected to evaluate how trustworthy a control loop is and thus, enhance/ ease the decisions of the NOs/SPs regarding the control loops that will be selected. Although the design of this mechanism has already been studied and presented in this chapter, its evaluation depends on the control loop to be studied each time and is out of the scope of this dissertation.

7.2 Metrics

Independent of the particular method (e.g. Bayesian Networks, Markov model) that may be applied for evaluating whether a control loop is trustworthy or not, a first important step is to define observable parameters/metrics that can be measured and monitored in a system, and that can be linked to a level of trustworthiness. A set of generic observable parameters that can be considered for the self-evaluation of control loops so as to derive a measure of their trustworthiness includes (but is not limited to) the following:

- Deviation from requested goals of a control loop (e.g., QoS levels)
- Resources involved for the enforcement of certain control loop(s) decisions/actions.
- Time required for the enforcement of control loop decisions/actions.
- Number of reconfigurations deriving from certain control loop decisions/actions.

It should be noted that different observable parameters may be considered for diverse self-management functionalities (control loops). Such observable parameters can be measured for each executed control loop to obtain an estimation of an "instantaneous trust index" [1], e.g., as a weighted sum of the observable parameters following an approach of [2] for the evaluation of an interaction between, say an agent and a user. It should also be noted that the term goal above refers to what should be achieved by a control loop. In order to achieve contractual agreement elements, self-management functions (control loops) must reach certain goals. Thus, if the decision of a certain control loop deviates from these goals this decision should not be deemed as trustworthy. Consequently, the larger the measured deviation, the number/amount of resources involved, the time required and the number of reconfigurations, the higher the level of inefficiency of certain control loop decisions/actions. In general a high level of inefficiency can be mapped to a low level of trustworthiness, i.e., to a low instantaneous trust index. More specifically, if the level of inefficiency of a control loop exceeds a specific predefined threshold, its instantaneous trust index will be decreased.

In order to obtain an "overall trust index", the long term performance of governance and self-management functions should also be taken into account, considering instantaneous as well as past information. The metrics, the instantaneous and the overall trust index are combined in the below proposed model, following a similar to Q-learning approach in order to enhance the decision making process of a control loop in terms of trust.

7.3 Approach

The method described here as a generic one targets at enforcing a control loop with the knowledge if its decisions are trustworthy enough. This knowledge may be exploited so as to improve the performance of the control loop by excluding decisions/actions that do not achieve desired trustworthiness levels. More specifically, the method stores information on situations encountered by the control loop including the corresponding decision that was applied for handling them in a knowledge base similar to the one depicted in Figure 7.1. This allows considering past interactions so as to allow faster and more efficient handling of problems. Thus, a control loop is enabled to reach its most trustworthy decisions given the current context of the system. Furthermore, the decision making process can be enhanced in terms of reduced time required for selecting a particular action, as a former "trustworthy" decision may be applied for the same/similar context without the need of executing an optimisation process.

Each time a decision is made, the control loop provides information on the contextual situation (parameters of the trigger for the Control loop) and the corresponding decision made. The values of relevant metrics are retrieved after the application of the decision of the control loop. The retrieved values are used so as to calculate and update the efficiency level, the instantaneous trust index and the overall trust index, given the context of the system. The knowledge-base is updated accordingly.

The proposed model follows Q-learning approach and the flow of Figure 7.2. In particular, after the selection of the metrics and the collection of the measurements, the latter will be used for calculating the level of efficiency of the control loop. The way that the Efficiency Level (EL) can be calculated is given by equation (7.1) and may involve different functions F such as the weighted sum of the metrics.

$$EL(t) = F(m_1, m_2, ..., m_n) \tag{7.1}$$

where $m_1, m_2, ..., m_n$ stand for the different metrics.

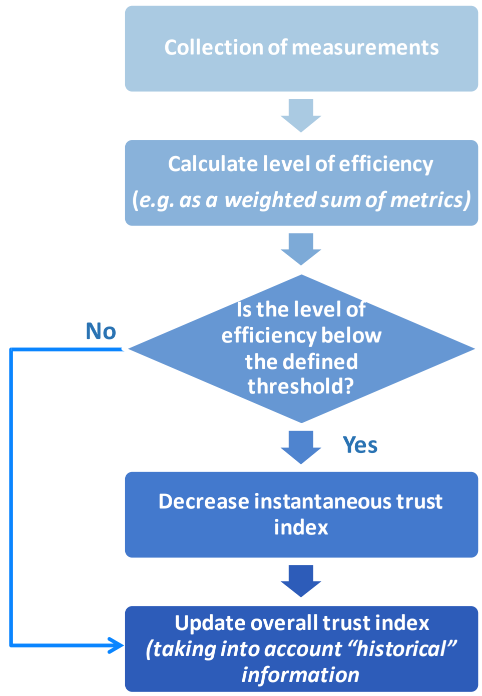


Figure 7.1: High-level view of control loop trustworthiness assessment process and update of corresponding knowledge base.

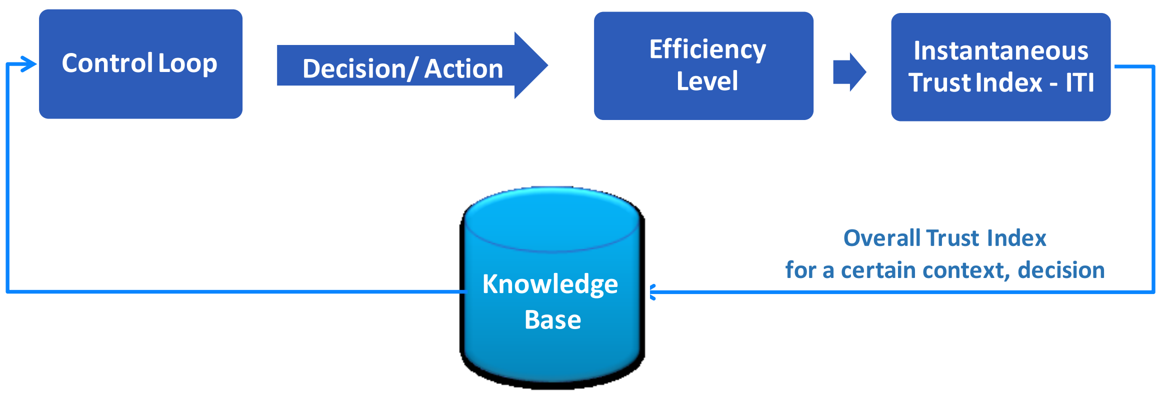
Additionally, a threshold for EL is defined with respect to equation (7.2), i.e., the value of function F when the minimum desired values of the metrics are used.

$$EL_{thres} = F(m_{1,opt}, m_{2,opt}, ..., m_{n,opt}) \tag{7.2}$$

Consequently, if the EL of the control loop decision is below the defined threshold, the control loop decision will be rated negatively, i.e., the Instantaneous Trust Index (ITI) r (corresponding to the payoff of Q-Learning technique) for the certain state s and action (decision)/ selected control loop a will be a negative real number. On the contrary, if the EL of the control loop decision is over the defined threshold, then the control loop decision will be rated positively, i.e., the corresponding payoff (ITI) r for the certain state s and action (decision) a will be a positive real number (equation 7.3).

$$EL(t)\begin{cases} < EL_{thres}, & r(s(t),a(t)) < 0 \\ > EL_{thres}, & r(s(t),a(t)) > 0 \end{cases} \tag{7.3}$$

where r(s(t),a(t)) represents the payoff for a particular system state s and action a at instance t.



State ID	Decision ID	Overall Trust Index - OTI
201111001	201111101	60%
201111002	201111201	75%
201111002	201111201	52%

Figure 7.2: Overview of process for updating the Overall Trust Index (OTI).

Accordingly, the ITIs will then be used to update the Overall Trust Index (OTI), which provides a more aggregated view of the trustworthiness of a particular decision/control loop over time (taking into account past trust estimations). In other words, it comprises both instantaneous information as well as "historical" information. Following the Q-learning approach [3], the OTI will first be calculated and then, during the next iterations, be updated until it reaches its maximum value according to equations (7.4) and (7.5):

$$Q(s(t),a(t)) = \left\langle \sum_{t=0}^{\infty} \gamma^t r(s(t),a(t)) \right\rangle_{s,r} \tag{7.4}$$

$$Q(s(t),a(t)) \rightarrow Q(s(t),a(t)) + \varepsilon[r(t) + \gamma \max_{a(t+1)} Q(s(t+1),a(t+1)) - Q(s(t),a(t))] \quad (7.5)$$

where Q(s(t),a(t)) corresponds to the OTI when the control loop is triggered by a situation (system state) s(t) and reaches decision a(t) and symbol $\langle \ \rangle_{s,r}$ refers to the average value. The discount factor 0<γ<1 stands for the weight of the payoff and is closely related to the time that has elapsed from the payoff, i.e. the larger γ designates that the more distant payoffs are more important. Moreover, parameter ε denotes the learning rate of the system.

As already mentioned, the derived knowledge on situations encountered and corresponding decision made by the control loop from the above described method will then be stored in a knowledge base. According to this knowledge base, the control loop will be enabled to select the most trustworthy decision given its inputs and the trigger. It should be noted that the focus of the presented method is more on assessing the performance of a control loop (or potentially a set of control loops) and consequently derive how much it can be trusted to operate autonomously and is not relevant to security. Nevertheless, the use of the described method and its outcomes may help to identify situations when conventional isolation mechanisms need to be applied.

7.4 Conclusions

Despite the many advantages that self-x mechanisms promise to provide network management with, disadvantages also exist, one of those coming from questions such as "how much can an autonomic control loop be trusted that will make the most appropriate decision? Will it be at least as good as the one made a human? What is the optimal control loop to incorporate in a network management?" All these question are closely related to the trust that needs to be built in autonomic control loop so as them to actually be deployed. In fact, although there is much ongoing research with respect to cognitive management systems, they have not yet been deployed in large large.

In order to bridge the gap, trust to self-x mechanisms needs to be build. To this end, this last chapter of the dissertation designs a knowledge building mechanism that could be used for evaluating the performance in terms of trust of an autonomic control loop.

However, even though the trust issue was raised during the dissertation and a way to address such a challenge was considered as intresting to be provided, the evaluation of this mechanism is closely related to the under question control loop increasing the boundaries of this research so much that was considered as out of scope of this dissertation.

7.5 Chapter References

[1] H. Chan, A. Segal. B. Arnold, I. Whalley, "How Can We Trust an Autonomic System to Make the Best Decision?," Autonomic Computing, 2005. ICAC 2005. Proceedings. Second International Conference on , vol., no., pp.351-352, 13-16 June 2005

[2] Y. Wang, J. Vassileva, "Bayesian Network-Based Trust Model", The Int. Conf. On Web Intelligence (WI'03), Halifax, Canada, 2003

[3] R. S. Sutton, A. G. Barto, "Reinforcement Learning An Introduction", Trends in Cognitive Sciences , Volume 9, Issue 5, MIT Press, 1998, DOI: 10.1109/TNN.1998.712192

8 SUMMARY – ONGOING CHALLENGES

This dissertation identified the need for mechanisms that can exploit Big Data and build knowledge that supports and enhances the decision making in CRSs. In particular, the increasing demand for network resources, the heterogeneity of the user devices and applications and the increased availability of digital data pose the challenges of future networks but also offer solutions for addressing them.

To this end, the dissertation presents the GANA architecture and identifies the placeholder where knowledge building mechanisms can be hosted so as to offer support to the envisaged functionality of the architecture and allow proactive diagnosis of future networks. Data parameters that can be monitored directly by the network have also been identified. Moreover, knowledge building mechanisms for a) estimating the network capabilities in terms of bitrate, b) predicting the congestion levels of a core link, c) foreseeing the traffic of a node, d) supporting and enhancing congestion avoidance mechanisms and e) building trust in autonomic control loops were designed, implemented and validated. The results from the validation of the mechanisms seem to be quite promising, although there is always room for improvement.

Last, but not least, despite the promising results, the concerns raised with respect to the trust in chapter 7 should not be underestimated. Research may propose enough autonomic control loops/ self-x mechanisms, but in order to move forward with their deployment, trust needs to be built.

9 APPENDIX A – ΕΚΤΕΤΑΜΕΝΗ ΕΛΛΗΝΙΚΗ ΠΕΡΙΛΗΨΗ

9.1 Περίγραμμα του Κεφαλαίου

Το κεφάλαιο αυτό αποτελεί μια σύνοψη της διατριβής στα ελληνικά. Έτσι, αρχικά παρουσιάζεται το πλαίσιο στο οποίο μπορούν να φιλοξενηθούν μηχανισμοί δημιουργίας γνώσης και τα δεδομένα που μπορούν να αξιοποιηθούν για την δημιουργία της γνώσης (κεφάλαιο 9.3). Ακολουθεί η ανάλυση μηχανισμών που βασίζονται σε τεχνικές μηχανικής μάθησης (machine learning) για την δημιουργία γνώσης που μπορεί να βελτιώσει την απόδοση των δικτύων σε ταχύτητα και ορθότητα των αποφάσεων που λαμβάνει. Συγκεκριμένα, το κεφάλαιο 9.4 δίνει έμφαση σε εκείνους τους μηχανισμούς που αποσκοπούν στην πρόβλεψη των δυνατοτήτων του και σε συγκριτική μελέτη μεταξύ διαφόρων τεχνικών μηχανικής μάθησης. Οι μηχανισμοί δημιουργίας γνώσης που αφορούν την πρόβλεψη του φορτίου αναλύονται στο κεφάλαιο 9.5. Τέλος, τα κεφάλαια 9.6 και 9.7 παρουσιάζουν αντίστοιχα πώς μηχανισμοί δημιουργίας γνώσης μπορούν να στηρίξουν το δίκτυο σε επίπεδο μεταφοράς και στην αυτο-αξιολόγηση των αποφάσεών του για την διαχείριση της εμπιστοσύνης προς αυτές.

9.2 Εισαγωγή

Η αυξανόμενη χρήση του Διαδικτύου αλλά και η βελτίωση των υπηρεσιών κατά τέτοιο τρόπο ώστε να προσφέρουν όλο και περισσότερες δυνατότητες στους χρήστες προκαλούν την αντίστοιχη αύξηση της ανάγκης για βελτιωμένη διαθεσιμότητα πόρων και επιπέδου χρήσης υπό την έννοια της παροχής αξιόπιστων και ποιοτικών υπηρεσιών προς τον τελικό χρήστη, αφήνοντας ελάχιστα αποδεκτά όρια Ποιότητας Υπηρεσίας (QoS) ή Ποιότητας Εμπειρίας Χρήσης (QoE). Τέτοιες απαιτήσεις, δεν μπορεί παρά να ασκούν περαιτέρω πίεση τους εκάστοτε δικτυακούς πόρους (διαθέσιμο εύρος φάσματος-bandwidth, επεξεργαστική ικανότητα σε δίκτυα κορμού και πρόσβασης).

Επιπροσθέτως, τόσο η όλο και πιο έντονη εισροή του Διαδικτύου στην καθημερινότητα του μέσου ανθρώπου/ χρήστη, είτε για την ενημέρωσή του είτε για την παροχή υπηρεσιών και δικτύωσης γενικότερα, όσο και η ταυτόχρονη βελτίωση και αναβάθμιση των τερματικών σταθμών οδηγούν στην εκθετική αύξηση των παραγόμενων δεδομένων που χρειάζεται να διαχειριστούν οι λειτουργοί των δικτύων (NOs) ή/ και οι πάροχοι των υπηρεσιών (SPs). Καθώς η διαχείριση, η αξιοποίηση και η εκμετάλλευση τόσων δεδομένων δεν είναι εφικτό να πραγματοποιηθεί μόνο από τον άνθρωπο, καθίσταται απαραίτητη η ανεύρεση αυτοματοποιημένων διαδικασιών προς αυτήν την κατεύθυνση. Οι διαδικασίες αυτές θα βοηθήσουν στην σωστή εκμετάλλευση της πληροφορίας που παράγεται και στην ανάλυση των "μεγάλων" - σε ποσότητα, ταχύτητα αλλαγής και διαφορετικότητα - δεδομένων (Big data analytics) και ομαδοποίησής τους κατά τρόπο εφικτό και άμεσα αξιοποιήσιμο από τον άνθρωπο, π.χ. «συναγερμοί» καταστάσεων στις οποίες θα επέλθει το δίκτυο στο άμεσο μέλλον (alarms), καθιστώντας δυνατή την γνωσιακή διαχείριση (cognitive management).

Στα συστήματα γνωσιακής διαχείρισης, η πρότερη εμπειρία του δικτύου καθοδηγεί την αυτοματοποιημένη λήψη αποφάσεων (decision making) του δικτύου για την βελτιστοποίηση της λειτουργίας του. Για τον λόγο αυτό χρησιμοποιούνται τεχνικές μηχανικής μάθησης (machine learning), οι οποίες εμπλουτίζουν τα δίκτυα με δυνατότητες επεξεργασίας της «πρωτοβάθμιας» πληροφορίας που λαμβάνουν από το περιβάλλον τους (sensing) ώστε να παράγουν υψηλότερου επιπέδου δεδομένα (elaborated data, alarms) και να δημιουργούν γνώση από την εμπειρία του δικτύου (knowledge building). Τα νέα αυτά επεξεργασμένα δεδομένα καθιστούν δυνατή την καλύτερη λήψη αποφάσεων είτε από τον διαχειριστή του δικτύου είτε ακόμα και αυτόνομα από το δίκτυο. Τέλος, αυτή η δυνατότητα των δικτύων μπορεί να αξιοποιηθεί είτε για την καθύστερη αντιμετώπιση προβλημάτων του δικτύου μετά την έγκυρη διάγνωσή τους (reactive diagnosis) ή ακόμα και ως πρόληψη για την αποφυγή ανεπιθύμητων καταστάσεων στο δίκτυο (proactive diagnosis).

Συνοψίζοντας, η γνωσιακή διαχείριση των δικτύων μπορεί να δώσει λύσεις στις προκλήσεις των μελλοντικών δικτύων αλλά αυτό προϋποθέτει την εισαγωγή γνώσης μέσα στο δίκτυο, όχι όμως γνώσης που παράγεται από τον άνθρωπο. Πρόκειται για γνώση παραγόμενη δυναμικά από το ίδιο το δίκτυο βάσει του περιβάλλοντος (context) μέσα στο οποίο ενεργεί, τα δεδομένα μεγάλου όγκου που παράγονται και την εμπειρία που έχει αποκτήσει από την μέχρι τώρα λειτουργία του. Η γνώση αυτή για να αποκτηθεί απαιτεί από το δίκτυο να εκπαιδεύεται δυναμικά από τον τρόπο που έχει αντιδράσει στο παρελθόν δεδομένου του περιβάλλοντος στο οποίο ενεργεί (context) και να είναι σε θέση να αναγνωρίσει ποιες από τις αντιδράσεις/ αποφάσεις αυτές το οδήγησαν σε καλύτερη λειτουργική κατάσταση ώστε την δεδομένη στιγμή να λαμβάνει τις αντίστοιχες αποφάσεις. Φυσικά, οι αποφάσεις αυτές θα διέπονται και από τους στόχους/ κανόνες (policies) του λειτουργού του δικτύου (NO) ή/ και του παρόχου της υπηρεσίας (SP). Ακολουθώντας το παραπάνω κύκλο λειτουργίας/ εργασιών, το δίκτυο δύναται να αποκτήσει μία δική του δυναμική και αυτοτέλεια ακόμη και ως προς την λήψη αποφάσεων.

Η διδακτορική διατριβή εστιάζει στην περιοχή των μηχανισμών δημιουργίας γνώσης. Συγκεκριμένα, μηχανισμοί μηχανικής μάθησης εξετάζονται και χρησιμοποιούνται για την ανάλυση δεδομένων μεγάλης κλίμακας που περιγράφουν παρελθούσες καταστάσεις του δικτύου και την συμπεριφορά του σύμφωνα με τις αποφάσεις που ελήφθησαν σε κάθε περίπτωση. Η ανάλυση των παραπάνω δεδομένων οδηγεί στην δημιουργία της γνώσης που μπορεί να αξιοποιηθεί για την πρόβλεψη ανεπιθύμητων μελλοντικών καταστάσεων του δικτύου και την βελτίωση των αποφάσεων είτε ως προς την αποτελεσματικότητά τους είτε ως προς την ταχύτητα λήψης τους.

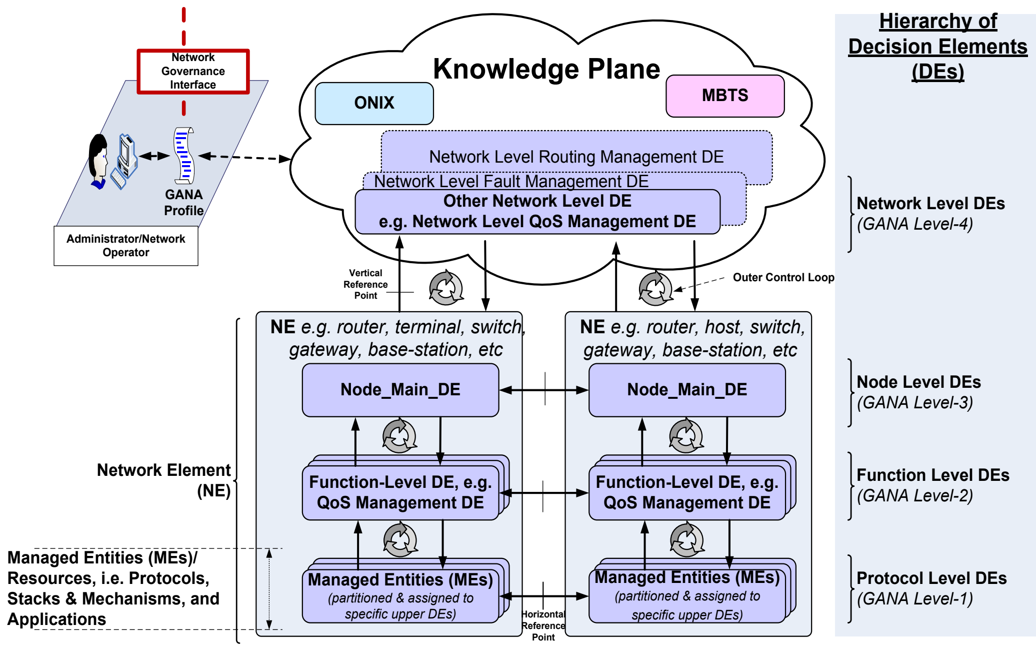
Εν γένει, οι τεχνικές μηχανικής μάθησης διαχωρίζονται στις καθοδηγούμενες τεχνικές εκπαίδευσης (supervised learning techniques) και στις μη καθοδηγούμενες (unsupervised learning techniques). Οι καθοδηγούμενες τεχνικές εκπαίδευσης είναι τεχνικές που λειτουργούν υπό επίβλεψη, τεχνικές δηλαδή στις οποίες το σύνολο των δεδομένων που εισάγονται κατά την εκπαίδευση περιέχει και το επιθυμητό αποτέλεσμα. Από αυτό το σύνολο των δεδομένων το σύστημα/ δίκτυο «ανακαλύπτει» την συνάρτηση που διέπει τα δεδομένα και καλείται να την γενικεύσει ώστε να μπορεί να βγάλει συμπεράσματα ακόμη και για δεδομένα άγνωστα προς αυτό, δεδομένα δηλαδή που δεν ανήκαν στο σύνολο των δεδομένων που εισήχθησαν κατά την εκπαίδευση. Ενδεικτικά αναφέρουμε ότι σε αυτού του είδους τις τεχνικές ανήκουν τα Νευρωνικά Δίκτυα (Neural Networks - NNs) [1]-[5], τα Bayesian δίκτυα [6]-[8] και οι τεχνικές βασισμένες σε Fuzzy-logic [9][10]. Μελέτες που έχουν λάβει χώρο σε ερευνητικό επίπεδο κάνοντας χρήση αυτών των τεχνικών για να προκύψουν οι προτιμήσεις των χρηστών του δικτύου και οι δυνατότητες του δικτύου είναι οι [7][8] και οι [1]-[6],[9],[10], αντίστοιχα.

Μη καθοδηγούμενες τεχνικές εκπαίδευσης είναι εκείνες οι τεχνικές οι οποίες προσπαθούν να συμπεράνουν την κρυφή δομή των δεδομένων χωρίς ωστόσο να τους γνωστοποιείται το επιθυμητό αποτέλεσμα. Τέτοιου είδους τεχνικές χρησιμοποιούνται ευρέως σε προβλήματα «εξόρυξης δεδομένων» (data mining) και ομαδοποίησης (clustering), ενώ ενδεικτικά σε αυτήν την κατηγορία βρίσκονται οι «κ-κοντινότεροι γείτονες» (k-nearest neighbors), τα «νευρωνικά αέρια» (neural gas), η «θεωρία προσαρμοστικής απήχησης» (Adaptive Resonance Theory - ART) [11], οι «χάρτες αυτό-οργάνωσης» (Self-Organizing Maps - SOMs) και παραλλαγές τους.

Στους μηχανισμούς που αναλύονται παρακάτω, έμφαση δόθηκε στις μη καθοδηγούμενες τεχνικές εκπαίδευσης, και ειδικά στους χάρτες αυτό-οργάνωσης και κάποιες από τις παραλλαγές τους ώστε να κατασκευαστούν αντίστοιχοι προς τα δεδομένα χάρτες και να αναγνωριστούν τα σχέδια/ μοντέλα στα οποία υπόκεινται οι μεταξύ τους σχέσεις. Τα μοντέλα αυτά αξιοποιήθηκαν εν συνεχεία για την πρόβλεψη/ προληπτική διάγνωση καταστάσεων στα οποία θα επέλθει το δίκτυο ή για την διευκόλυνση της διαδικασίας αυτόματης λήψης αποφάσεων από το ίδιο το δίκτυο.

9.3 Δομές και Μοντέλα Απεικόνισης Γνωσης

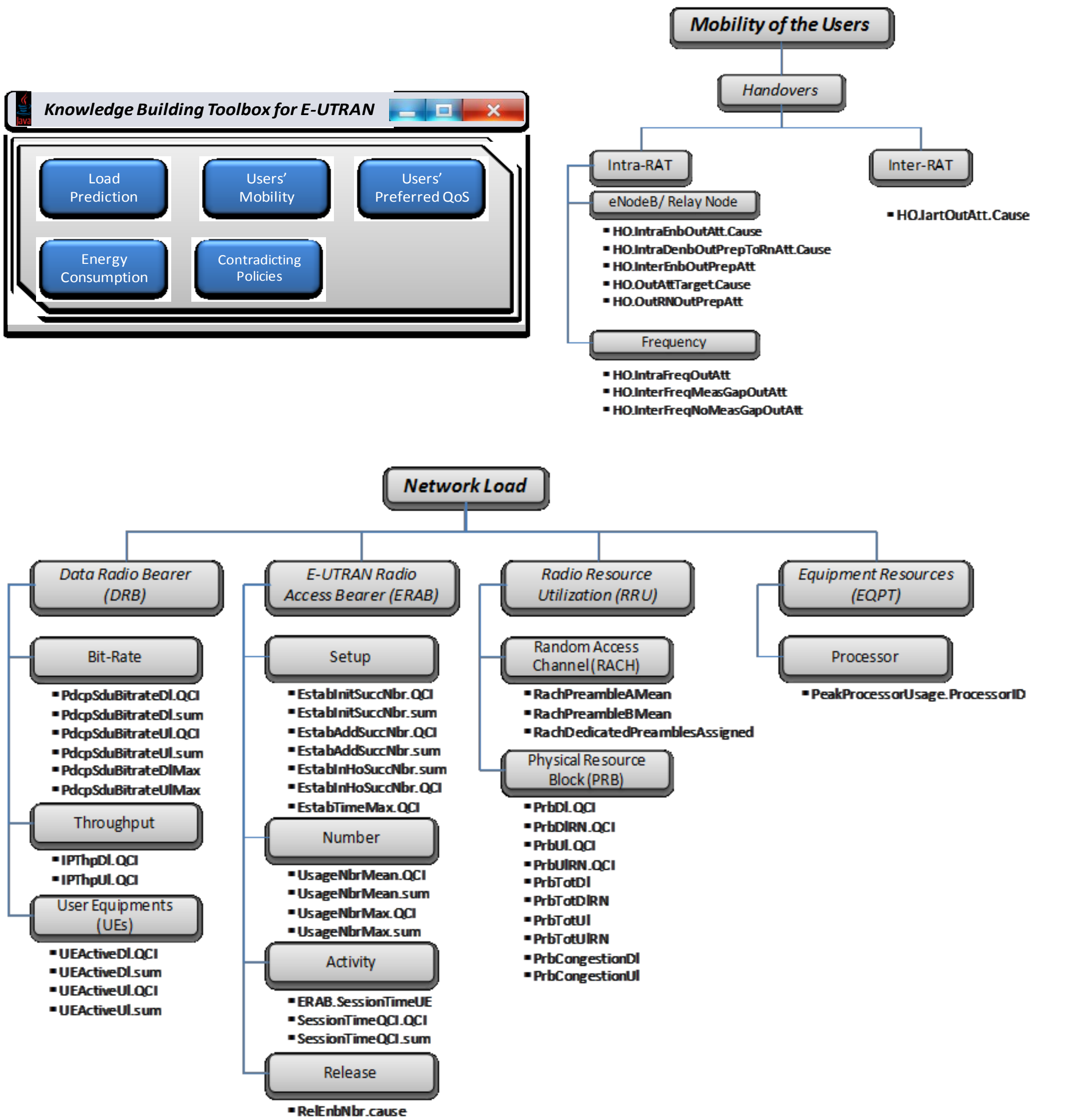
Η αρχιτεκτονική GANA [12] έχει προταθεί από την ομάδα προτυποποίησης ETSI GS AFI και αφορά στην περιγραφή μιας αρχιτεκτονικής που θα μπορεί να περιγράψει τις λειτουργίες των μελλοντικών δικτύων αλλά και την οργάνωση των επιμέρους επιπέδων των στοιχείων του δικτύου. Σε αυτήν προβλέπεται η ύπαρξη του "Knowledge Plane", το οποίο ευθύνεται για την μετάφραση των GANA profiles, των οδηγιών δηλαδή που εισάγει ο λειτουργός του δικτύου, με οδηγίες που μπορούν να εκτελεστούν από μηχανές αποφάσεων χαμηλότερων επιπέδων στην αρχιτεκτονική. Στα πλαίσια της δημοσίευσης [13] προτάθηκε η εισαγωγή στο "Knowledge Plane" του "Knowledge Functional Block", δηλαδή μιας ομάδας διεργασιών/μηχανισμών που θα συμπληρώνει την πληροφορία σχετικά με τις πολιτικές χρήσης και διαχείρισης των δικτύων και τους high-level στόχους των λειτουργών των δικτύων, όπως αυτές προδιαγράφονται στην αρχιτεκτονική GANA (Εικόνα 9.1) [12], με την "εμπειρία" του δικτύου από παλαιότερες μετρήσεις και παρατηρήσεις. Κάθε μηχανισμός του "Knowledge Functional Block" αποσκοπεί στην παροχή διαφορετικής πληροφορίας σχετικά με τον τρόπο λειτουργίας του δικτύου ενώ για την παραγωγή αυτής της γνώσης μπορούν να χρησιμοποιηθούν διαφορετικές τεχνικές μηχανικής μάθησης (ανάλογα με την φύση του προβλήματος που αντιμετωπίζεται) και να συνδυαστούν πληροφορίες διαφορετικής φύσης όπως πχ., μετρήσεις του δικτύου με την ημερομηνία, τις επιλογές των χρηστών, περιβαλλοντικά δεδομένα κ.α.

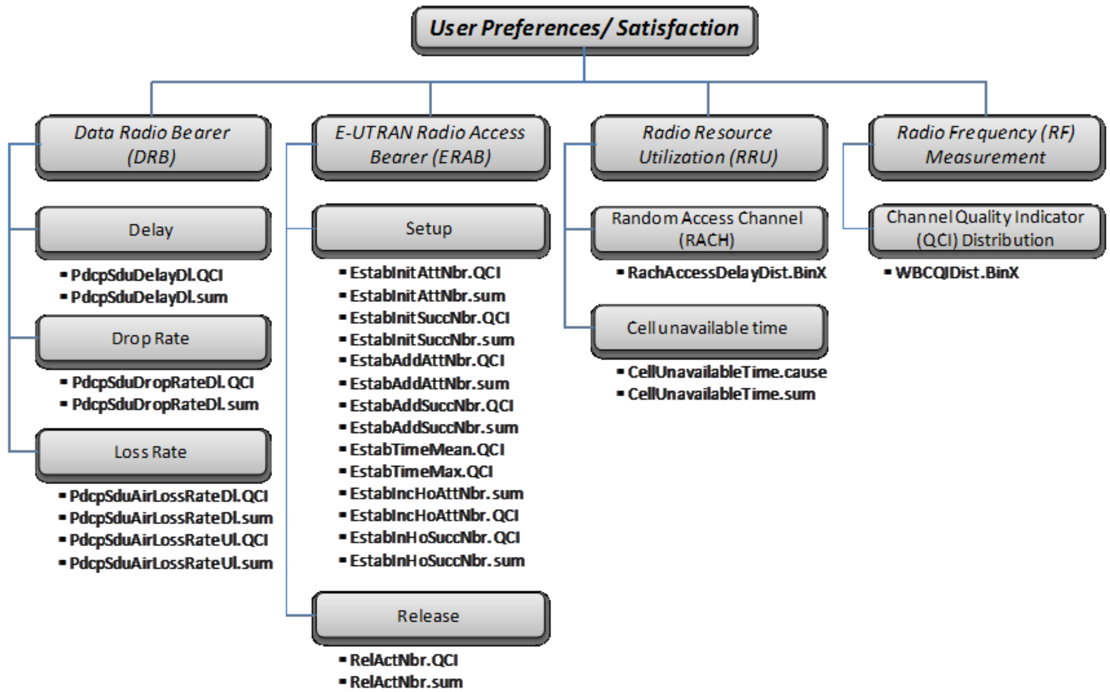


Εικόνα 9.1: Αρχιτεκτονική GANA [12].

Η Εικόνα 9.2 απεικονίζει το γραφικό περιβάλλον που προβλέπεται για το knowledge functional block ώστε να μπορεί ο λειτουργός του δικτύου να παρακολουθεί την γνώση που δημιουργείται από τους αντίστοιχους μηχανισμούς γνώσης. Για τον προσδιορισμό των δεδομένων που μπορούν να χρησιμοποιηθούν από τους παραπάνω μηχανισμούς ώστε να δημιουργηθεί η επιθυμητή γνώση στην περίπτωση των δικτύων 4ης γενιάς (LTE) [14] μελετήθηκαν τα έγγραφα προτυποποίησης του «Έργου Κοινοπραξίας Τρίτης Γενιάς» (3GPP). Η Εικόνα 9.2 παρουσιάζει τις δομές των δεδομένων που μπορούν να αντληθούν άμεσα από βάσεις διαχείρισης πληροφορίας (MIBs) των δικτύων 4ης γενιάς και να αξιοποιηθούν για την δημιουργία γνώσης σε 3 ενδεικτικά παραδείγματα: α) την πρόβλεψη της κινητικότητας των χρηστών, β) την πρόβλεψη του φορτίου του δικτύου και γ) την πρόβλεψη των επιλογών του χρήστη σύμφωνα με την μελέτη [13].

Οι μηχανισμοί μάθησης και διαχείρισης γνώσης του "Knowledge Functional Block" αλληλεπιδρούν με τους μηχανισμούς αίσθησης και επίγνωσης (sensing and awareness) για να λαμβάνουν τα πρωτογενή δεδομένα του δικτύου (raw data) και τους μηχανισμούς αποφάσεων (decision making mechanisms) στους οποίους παραδίδουν την γνώση για την βελτίωση και διευκόλυνση των αποφάσεών τους [15].





Εικόνα 9.2: Γραφικό περιβάλλον του knowledge functional block και παράμετροι του δικτύου που μπορούν να χρησιμοποιηθούν για ενδεικτικά παραδείγματα μηχανισμών γνώσης [13].

9.4 Μηχανισμοί Ανάπτυξης Γνώσης

Σε αυτό το κεφάλαιο παρουσιάζεται μιας εργαλειοθήκη που φιλοξενεί περαιτέρω του ενός μηχανισμού παραγωγής γνώσης με σκοπό την βελτίωση της διαχείρισης του δικτύου. Για την δημιουργία αυτής της εργαλειοθήκης μελετήθηκαν διαφορετικά σενάρια και συνθήκες υπό τις οποίες μια τέτοια εργαλειοθήκη θα μπορούσε να βελτιώσει τις αποφάσεις της συσκευής που την φέρει, όπως η στηριγμένη στην γνώση διαχείριση αναδιαρθρώσιμων (reconfigurable) συσκευών, η αυτό-βελτιστοποίηση γνωσιακών συσκευών χρήστη και η διαχείριση γνωσιακών συστημάτων και ευκαιριακών δικτύων (opportunistic networks).

Οι μηχανισμοί που συμπεριλήφθηκαν σε αυτήν την εργαλειοθήκη περιελάμβαναν μηχανισμούς που μπορούν να αποκτήσουν και να μάθουν πληροφορίες για τον χρήστη (όπως πληροφορίες για τις προτιμήσεις του, την συμπεριφορά του αλλά και τις δυνατότητες της συσκευής του), πληροφορίες για τις συνθήκες κάτω από τις οποίες λειτουργεί το δίκτυο ή η συσκευή του χρήστη (context information), την πολιτική/ κανονισμούς (policies) σύμφωνα με τους οποίους οφείλει να συνεργαστεί και την αποδοτικότητα των αποφάσεών του. Πληροφορίες που σχετίζονται με τις συνθήκες λειτουργίας του δικτύου είναι η κατάσταση των στοιχείων του δικτύου (π.χ., των σταθμών βάσης), η κατάσταση του περιβάλλοντός του και οι συσκευές των χρηστών. Αντίστοιχα, οι πληροφορίες αυτές από την πλευρά της συσκευής του χρήστη μεταφράζονται σε δεδομένα σχετικά με τις τεχνολογίες πρόσβασης σε μια συγκεκριμένη περιοχή και την αντίστοιχη κατάστασή τους (π.χ., συχνότητες που χρησιμοποιούνται, διαθέσιμοι πόροι και κάλυψη), πληροφορίες για την κατάσταση της συσκευής (π.χ., κάλυψη στην συγκεκριμένη περιοχή, διαθέσιμη μπαταρία και δυνατότητες που έχει βάσει της τεχνολογίες της), και πληροφορίες για άλλες συσκευές που λειτουργούν στην περιοχή (π.χ., δραστηριότητα και δυνατότητα συνεργασίας).

Τέλος, για τους μηχανισμούς αυτούς στα πλαίσια του [16] προτάθηκαν συγκεκριμένες αλγοριθμικές λύσεις (implementation approaches) βασισμένες σε διαφορετικές τεχνικές μάθησης.

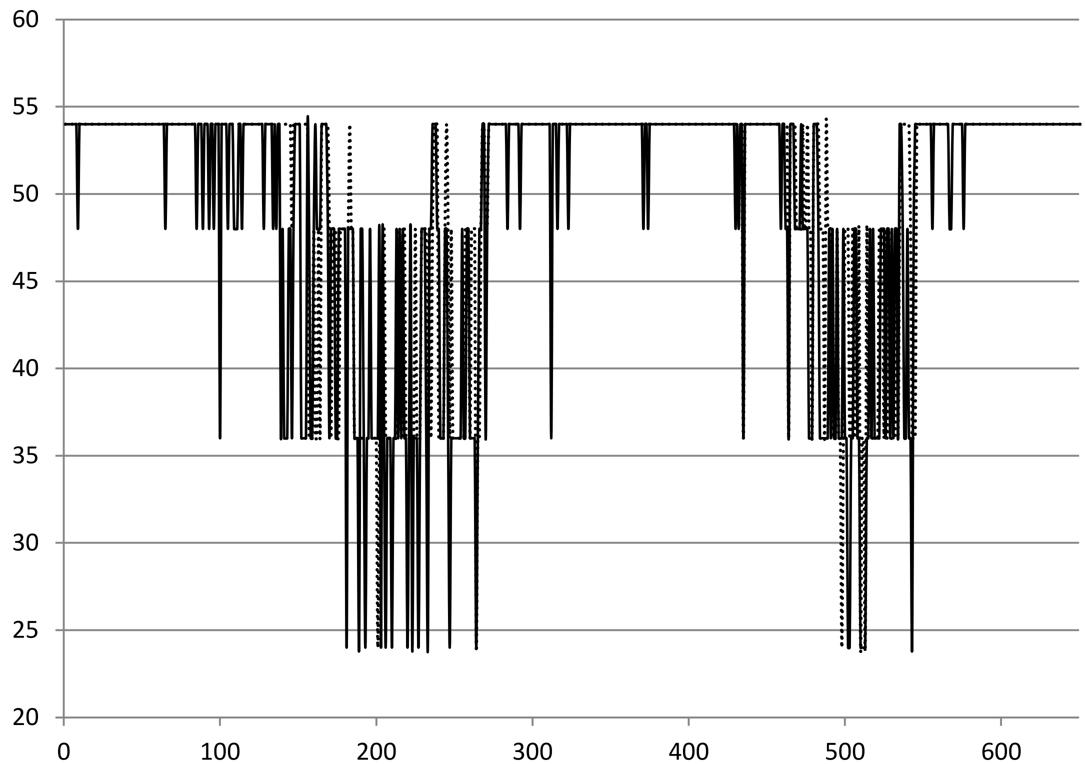
9.4.1 Δημιουργία γνώσης για την εκτίμηση των δυνατοτήτων του δικτύου - Συγκριτική μελέτη μεταξύ NN και SOMs

Ο μηχανισμός δημιουργίας γνώσης για την εκτίμηση των δυνατοτήτων του δικτύου χρησιμοποιεί δεδομένα όπως είναι ο θόρυβος, ο δείκτης έντασης ληφθέντος σήματος (RSSI), εισερχόμενα και εξερχόμενα λάθη, πακέτα και Bytes – ως αποτέλεσμα μιας διαμόρφωσης (configuration) του δικτύου – και τα συσχετίζει με το ρυθμό μετάδοσης (bitrate) – δείκτη ποιότητας του σήματος (QoS) – που μπορεί να επιτευχθεί όταν έχει επιλεγεί η συγκεκριμένη διαμόρφωση. Κάθε συνδυασμός των τιμών των παραμέτρων αυτών θα αναφέρεται στο εξής ως δείγμα δεδομένων.

Στον μηχανισμό αυτό χρησιμοποιήθηκε η τεχνική αυτό-οργανωμένων χαρτών (SOM). Η τεχνική αυτή προτάθηκε πρώτη φορά από τον T. Kohonen [17][18] και απεικονίζει πολυδιάστατα δεδομένα σε δισδιάστατους χάρτες. Αυτοί οι χάρτες αποτελούνται από τετραγωνικές ή εξαγωνικές κυψέλες οι οποίες βρίσκονται πάνω σε πλέγμα κανονικού συστήματος ενώ κάθε δείγμα δεδομένων σχετίζεται με εκείνη την κυψέλη/ νευρώνα του χάρτη που το θέτει ανάμεσα σε όσο το δυνατόν πιο συγγενικά (σε επίπεδο διανυσμάτων) δείγματα. Υπό αυτήν την έννοια, ο χάρτης που δημιουργείται αναπαριστά την ομοιότητα των δειγμάτων δεδομένων και την κατηγοριοποίησή τους και ως εκ τούτου μπορεί να χρησιμοποιηθεί και για την αναγνώριση των σχεδίων/ μοντέλων των δεδομένων (pattern recognition). Η δημιουργία της γνώσης βασίστηκε σε μετρήσεις που είχαν λάβει χώρα σε πραγματικό περιβάλλον εντός των ορίων του Πανεπιστημίου Πειραιώς ενώ η γνώση που παράχθηκε χρησιμοποιήθηκε εν συνεχεία από έναν μηχανισμό που μπορούσε να την αξιοποιήσει για την πρόβλεψη του bitrate τόσο γνωστών όσο και άγνωστων δειγμάτων δεδομένων.

Για την ακρίβεια, ο μηχανισμός που είχε αναπτυχθεί χρησιμοποιεί την ίδια λογική με την εκπαίδευση του χάρτη: κατ’ αρχήν, κάθε δείγμα δεδομένων αντιστοιχείται με μία κυψέλη χρησιμοποιώντας τις ίδιες παραμέτρους με αυτές που χρησιμοποιήθηκαν κατά την διάρκεια της εκπαίδευσης του χάρτη και στην συνέχεια γίνεται η εκτίμηση του ρυθμού μετάδοσης (bitrate) ανάλογα με το ρυθμό μετάδοσης που παρατηρείται στα πλησιέστερα επάνω στον χάρτη «εκπαιδευτικά» δείγματα δεδομένων.

Τελικά, ο μηχανισμός αξιολογήθηκε συγκρίνοντας τις τιμές του ρυθμού μετάδοσης των δειγμάτων δεδομένων που προέκυψαν από τον μηχανισμό με τις πραγματικές μετρήσεις του ρυθμού μετάδοσης των ίδιων δειγμάτων δεδομένων. Μετά από αρκετές περιπτώσεις ελέγχου σχετικά με τον καταλληλότερο συνδυασμό παραμέτρων ο ρυθμός μετάδοσης είχε προβλεφθεί σωστά σε ποσοστό 78,9% των δειγμάτων δεδομένων που είχαν εξεταστεί χρησιμοποιώντας α) τον αριθμό εισερχόμενων πακέτων, β) τον αριθμό των εξερχόμενων πακέτων και γ) τον δείκτη έντασης ληφθέντος σήματος. Το συγκριτικό διάγραμμα των προβλεφθέντων και των αντίστοιχων πραγματικών τιμών του ρυθμού μετάδοσης σε αυτήν την περίπτωση φαίνονται στην Εικόνα 9.3.



Εικόνα 9.3: Συγκριτικό διάγραμμα των προβλεφθέντων (συνεχής γραμμή) και των αντίστοιχων πραγματικών (διακεκομμένη γραμμή) τιμών του ρυθμού μετάδοσης.

Συγκρίνοντας τον μηχανισμό με αντίστοιχους μηχανισμούς της βιβλιογραφίας που βασίζεται στην δημιουργία της ίδιας γνώσης χρησιμοποιώντας τα νευρωνικά δίκτυα [1][2] προέκυψαν τα εξής αποτελέσματα [19]: Παρά το γεγονός ότι σε κάποιες περιπτώσεις ο μηχανισμός που στηρίζεται στα νευρωνικά δίκτυα παρουσιάζει καλύτερα αποτελέσματα (μικρότερο μέσο τετραγωνικό σφάλμα μεταξύ των προβλεφθέντων και των πραγματικών τιμών του ρυθμού μετάδοσης) από αυτών της τεχνικής SOM, ο τελευταίος χρειάζεται λιγότερα ιστορικά δεδομένα και μικρότερης διάρκειας εκπαίδευση ενώ παράλληλα προσφέρει την ευελιξία εύκολης και γρήγορης αλλαγής του αριθμού και του τύπου των παραμέτρων που χρησιμοποιούνται χωρίς να απαιτεί τον επανασχεδιασμό του μηχανισμού.

9.4.2 Δημιουργία γνώσης για την εκτίμηση των δυνατοτήτων του δικτύου - Συγκριτική μελέτη μεταξύ Bayesian Statistics και SOMs

Μηχανισμοί που αφορούν την διάγνωση των δυνατοτήτων του δικτύου είτε από την πλευρά του δικτύου είτε από την πλευρά της συσκευής του χρήστη και την μελέτη των προτιμήσεων του χρήστη στηριζόμενοι στην στατιστική Bayesian έχουν επίσης δημιουργηθεί και αναφερθεί στην βιβλιογραφία. Οι μελέτες [6][7] και [8] είναι μερικά τέτοια παραδείγματα. Ο SOM-based μηχανισμός που περιγράφηκε στο κεφάλαιο 9.4.1 συγκρίθηκε με αυτούς τους μηχανισμούς και μελετήθηκε πώς αυτοί μπορούν να συνδυαστούν για να προσφέρουν ακόμη καλύτερη αποτελεσματικότητα.

Συγκεκριμένα, ο SOM-based μηχανισμός για την δημιουργία γνώσης των δυνατότητες των δικτύων (ταχύτητα μετάδοσης bitrate) του [19] συγκρίνεται αφενός με έναν Bayesian-based όμοιό του που κάνει εκτίμηση των δυνατοτήτων του δικτύου και έναν Bayesian-based που αποσκοπεί στην δημιουργία γνώσης σχετικά με τις προτιμήσεις των χρηστών του δικτύου (πόσο ικανοποιημένοι έμειναν οι χρήστες από την απόδοση της εφαρμογή που μόλις χρησιμοποίησαν). Από την μελέτη που πραγματοποιήθηκε και παρουσιάστηκε στο [20] προέκυψαν τα εξής:

- Συγκρίνοντας τις δύο τεχνικές μάθησης (στατιστική Bayesian και αυτό-οργανωμένοι χάρτες) στο ίδιο πρόβλημα (την δημιουργία γνώσης σχετικά με τις δυνατότητες του δικτύου), παρατηρήθηκε ότι η τεχνική Bayesian, εν αντιθέσει με τους αυτό-οργανωμένους χάρτες διαθέτει επιπλέον την δυνατότητα της online εκπαίδευσης και δεν επαναχρησιμοποιεί δεδομένα που εξέτασε στο παρελθόν. Ως εκ τούτου, απαιτεί λιγότερες αποθηκευτικές δυνατότητες από την συσκευή που την φιλοξενεί και άρα μπορεί να είναι ακόμη και κομμάτι της συσκευής του χρήστη.
- Οι δύο τεχνικές που χρησιμοποιούνται για την μελέτη των δυνατοτήτων του δικτύου μπορούν να συνδυαστούν, η μία από την σκοπιά του δικτύου και η άλλη από την πλευρά της συσκευής του χρήστη, ώστε η τελευταία να επιβεβαιώνει τις δυνατότητες που δηλώνει το δίκτυο ότι διαθέτει.
- Τέλος, η γνώση που παράγεται σχετικά με τις επιθυμίες του χρήστη μπορεί να συνδυαστεί με την γνώση σχετικά με τις δυνατότητες του δικτύου και μαζί να αξιοποιηθούν από ένα μηχανισμό επιλογής δικτύου βάσει των προσφερόμενων

δυνατοτήτων και τις προτιμήσεις του χρήστη ώστε να αυξάνουν την ποιότητα εμπειρίας (QoE) του χρήστη και της υπηρεσίας (QoS) του δικτύου.

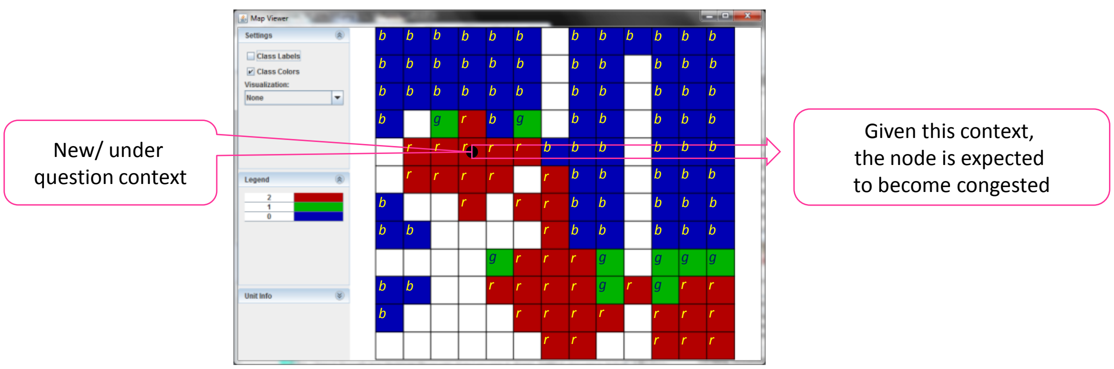
9.5 Γνωσιακή Ανάλυση Δεδομένων Ασύρματων Ευρυζωνικών Υποδομών Πρόσβασης

9.5.1 Πρόβλεψη του επιπέδου συμφόρησης μιας ζεύξη του κεντρικού μέρους δικτύου

Ο μηχανισμός αυτός αφορά την πρόβλεψη του επιπέδου συμφόρησης που θα παρατηρηθεί μέσα στο επόμενο χρονικό διάστημα πάνω σε μια ζεύξη του κεντρικού μέρους του δικτύου (core network). Η πρόβλεψη αυτή θα στηρίζεται σε παραμέτρους που μπορούν να παρατηρηθούν απευθείας από το δίκτυο. Κατόπιν μελέτης σχετικής βιβλιογραφίας επί του θέματος, οι μεταβλητές που εξετάσθηκαν σχετίζονται είτε με την κίνηση του δικτύου, σε επίπεδο εισερχόμενης στην ζεύξη κίνησης του μετρημένη σε Bytes ή/ και σε πακέτα και την αντίστοιχη τάση της κίνησης, είτε με χαρακτηριστικά του δικτύου και της ζεύξης όπως είναι το μέγεθος καταχωρητή του κόμβου που ελευθερώνει την κίνηση μέσα στην ζεύξη, η ουρά αναμονής του ίδιου κόμβου, η χωρητικότητα της ζεύξης και τα πακέτα που χάνονται.

Η αξία αυτής της μελέτης δεν έγκειται στην ανάλυση της σχέσης της καθεμιάς εκ των παραμέτρων αυτής καθ’ αυτής με την κίνηση, και κατ’ επέκταση την συμφόρηση, του δικτύου. Αντιθέτως, αυτές οι σχέσεις είναι λίγο-πολύ γνωστές. Η αξία αυτής της μελέτης οφείλεται στο γεγονός ότι αν και γνωρίζουμε την άμεση σχέση της κάθε παραμέτρου με την συμφόρηση του δικτύου, όταν παραπάνω από μία παραμέτρους αλλάξουν δεν είναι πάντα γνωστή η «αντίδραση» του δικτύου, δηλαδή το αν θα αυξηθεί, θα παραμείνει σταθερή ή ακόμα και αν θα μειωθεί η κίνηση του δικτύου. Πολύ δε περισσότερο, είναι δύσκολο να δώσουμε μια γρήγορη εκτίμηση του αν αυτή η αλλαγή της κίνησης θα είναι ικανή να φέρει την ζεύξη πολύ κοντά ή να την απομακρύνει τελείως από μια συμφόρηση.

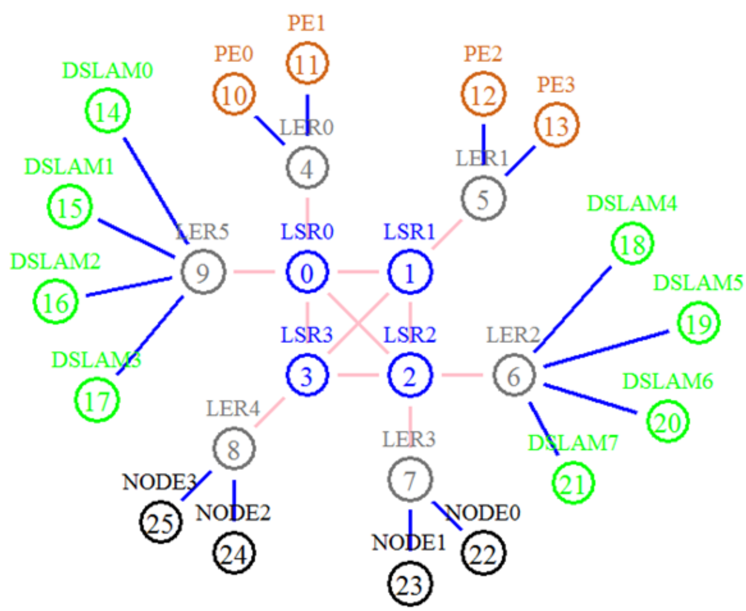
Ορμώμενοι από την παραπάνω παρατήρηση, αυτή η ερευνητική δραστηριότητα είχε σαν στόχο της να προσδιορίσει τη συμπεριφορά μιας ζεύξης σε όρους συμφόρηση, όταν αλλάζουν περαιτέρω της μίας παραμέτρου. Για τον σκοπό αυτό χρησιμοποιήθηκε η τεχνική εκπαίδευσης «Αυτό-οργανωμένοι Χάρτες» (SOMs), σύμφωνα με την οποία πολυδιάστατα δεδομένα μπορούν να απεικονιστούν πάνω σε δισδιάστατους χάρτες διατηρώντας την πληροφορία σχετικά με την μεταξύ τους απόσταση (διαφορά). Επιπροσθέτως, το ρόλο των πολυδιάστατων δεδομένων έπαιξαν σε αυτήν την περίπτωση οι συνθήκες μέσα στις οποίες λειτουργούσε η ζεύξη, δηλαδή ο συνδυασμός των δεδομένων που δίνουν πληροφορίες για την τιμή όλων των παραμέτρων που προαναφέρθηκαν. Κάθε τέτοιος συνδυασμός θα αναφέρεται από εδώ και στο εξής ως δείγμα δεδομένων. Σαν τελικό αποτέλεσμα, τα δεδομένα είναι οργανωμένα σε ομάδες (clusters), ανάλογες με την ομοιότητα των δεδομένων, πάνω στον χάρτη που παράχθηκε με την συγκεκριμένη τεχνική. Ως εκ τούτου, ο χάρτης απεικονίζει την σχέση που έχουν μεταξύ τους οι παράμετροι που χρησιμοποιήθηκαν με το επίπεδο συμφόρησης (Εικόνα 9.4).



Εικόνα 9.4: Επισκόπηση του μηχανισμού πρόβλεψης του επιπέδου συμφόρησης της ζεύξης.

Μετά την δημιουργία του χάρτη, η ανίχνευση του επιπέδου συμφόρησης για ένα νέο δείγμα δεδομένων είναι θέμα κατάταξης του δείγματος μέσα στην ομάδα που αντιστοιχεί. Ως εκ τούτου, και χρησιμοποιώντας την ίδια τεχνική, κάθε νέο δείγμα χαρτογραφείται επάνω στον χάρτη που έχει δημιουργηθεί. Αυτές οι συνθήκες λειτουργίας της ζεύξης, αντιστοιχούν σε συνθήκες τέτοιου επιπέδου συμφόρησης σαν και αυτό που αντιπροσωπεύει το μεγαλύτερο πλήθος δειγμάτων δεδομένων της ομάδας (cluster). Εάν τα περισσότερα δείγματα δεδομένων αντιστοιχούν σε συνθήκες λειτουργίας της ζεύξης που στο παρελθόν προκάλεσαν συμφόρηση τότε το νέο δείγμα δεδομένων αντιστοιχεί επίσης σε συνθήκες λειτουργίας της ζεύξης που θα προκαλέσουν συμφόρηση (κόκκινα – r – τετράγωνα του χάρτη). Εναλλακτικά, αν τα περισσότερα δείγματα δεδομένων αντιστοιχούν σε συνθήκες λειτουργίας μη κορεσμένης ζεύξης (μπλε – b – τετράγωνα του χάρτη) ή ζεύξης κοντά στην συμφόρηση (πράσινα – g – τετράγωνα του χάρτη), το νέο δείγμα δεδομένων αντιστοιχεί επίσης σε τέτοιες συνθήκες λειτουργίας.

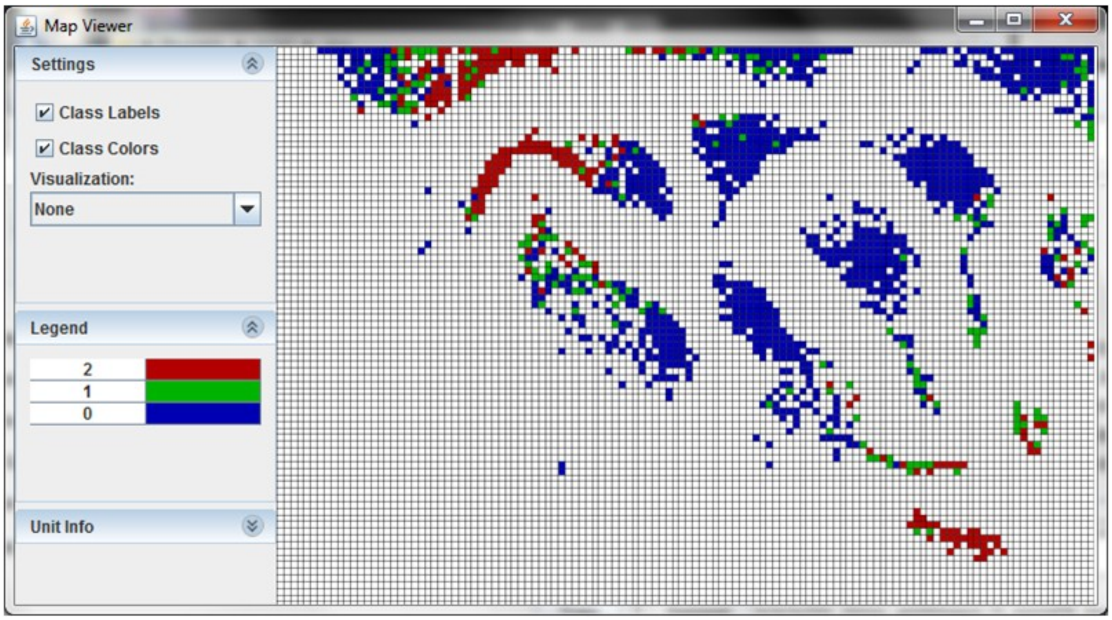
Για την δοκιμή της μεθόδου χρησιμοποιήθηκαν δεδομένα που δημιουργήθηκαν με την βοήθεια του NS2. Για την ακρίβεια προσομοιώθηκε η τοπολογία δικτύου της Εικόνα 9.5.



Εικόνα 9.5: Τοπολογία Δικτύου Προσομοίωσης.

Σε αυτήν την τοπολογία, η κύρια κίνηση προέρχεται από υπηρεσίες VoIP, TCP και UDP. Επιπλέον, η υπό εξέταση σχετικά με τη συμφόρηση ζεύξη είναι αυτή που συνδέει τους κόμβους 0 και 2 (LSR0, LSR2). Ως εκ τούτου τα δεδομένα που παρακολουθούνται αφορούν δεδομένα εισερχόμενα στον κόμβο 0 (LSR0) με προορισμό τον κόμβο 2 (LSR2), τα χαρακτηριστικά του κόμβου 0 ή στην ζεύξη αυτή καθ’ αυτή. Μετά από έναν αριθμό δοκιμών, που χρησιμοποιούσαν τα δεδομένα της προσομοίωσης, για την επιλογή των καταλληλότερων, από τις προαναφερθείσες, παραμέτρων, οι περισσότερες σωστές προβλέψεις, σε ποσοστό 86,6%, έγιναν όταν το δείγμα δεδομένων περιελάμβανε τις εξής παραμέτρους: α) αριθμός εισερχομένων Bytes, β) τάση εισερχομένων Bytes, γ) χωρητικότητα ζεύξης, δ) μέγεθος ουράς αναμονής, ε) μέγεθος καταχωρητή, στ) αναλογία χαμένων προς σταλμένων πακέτων. Ο χάρτης που δημιουργήθηκε από αυτά τα δεδομένα και χρησιμοποιήθηκε για τις προβλέψεις του επιπέδου συμφόρησης είναι αυτός της Εικόνα 9.6.

Περαιτέρω πληροφορίες για τον μηχανισμό μπορούν να αντληθούν από τα [21] και [22].



Εικόνα 9.6: Χάρτης που απεικονίζει την ομαδοποίηση των δεδομένων ανάλογα με το επίπεδο συμφόρησης στο οποίο οδηγούν (κόκκινο για συμφόρηση, μπλε για μη κορεσμένη ζεύξη και πράσινο για ζεύξη που είναι κοντά στην συμφόρηση, χάνει δηλαδή κάποια πακέτα αλλά δεν θεωρεί.

9.5.2 Πρόβλεψη της κίνησης του δικτύου

Ο μηχανισμός αυτός αφορά στην πρόβλεψη της μελλοντικής κίνησης του δικτύου σε μια συγκεκριμένη περιοχή κάνοντας χρήση δεδομένων που προέρχονται από ποικίλες πηγές, και δεν παρατηρούνται μόνο απευθείας στο δίκτυο (π.χ., το φορτίο του δικτύου). Αυτά τα δεδομένα έχουν πιο ανθρωποκεντρικό προσανατολισμό και μπορεί να αφορούν πληροφορία σχετικά με την ημερομηνία ή την θερμοκρασία ή το είδος της περιοχής (π.χ., κατοικημένη περιοχή ή περιοχή που περιλαμβάνει το πλείστον χώρους διασκέδασης). Η επιλογή αυτή είναι σε απόλυτη συμφωνία τόσο με την απαίτηση των μελλοντικών δικτύων για δυναμική διαχείριση των πόρων τους όσο και με την ανάγκη αξιοποίησης και διαχείρισης των συνεχώς αυξανόμενων ψηφιακών δεδομένων (Big Data). Ως εκ τούτο, η συμπεριφορά του μηχανισμού εξετάστηκε και προς τις δύο αυτές κατευθύνσεις. Ο μηχανισμός που αναπτύχθηκε βασίζεται στην τεχνική αυτο-οργανωμένων χαρτών (SOMs) και την αναγνώριση του σχεδίου που ακολουθεί η κίνηση του δικτύου (pattern recognition). Τρία διαφορετικά σενάρια που αξιοποιούν διαφορετικά δεδομένα εισόδου (input data) μελετήθηκαν:

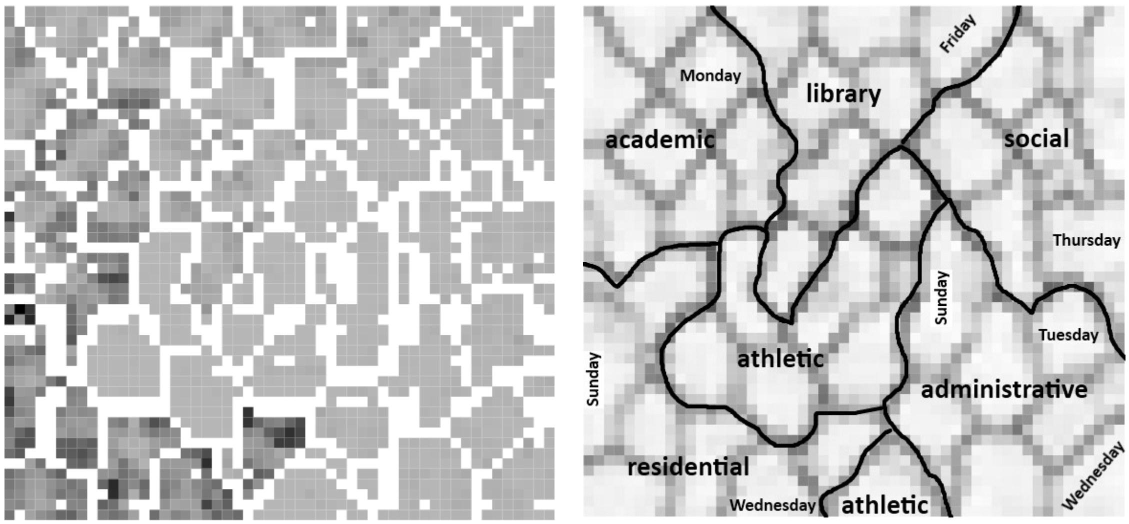
- Μελέτη του φορτίου ενός δικτύου σύμφωνα με την ώρα, την ημέρα, την περιοχή και τον καιρό στην περιοχή,
- Μελέτη του φορτίου ενός δικτύου σύμφωνα με την ώρα, την ημέρα, την περιοχή, την εβδομάδα του χρόνου και το αν η εν λόγω μέρα είναι αργία ή όχι και
- Μελέτη του φορτίου ενός δικτύου σύμφωνα με όλες τις παραπάνω παραμέτρους.

9.5.2.1 Μελέτη του φορτίου ενός δικτύου σύμφωνα με την ώρα, την ημέρα, την περιοχή και τον καιρό στην περιοχή.

Τα δεδομένα του δικτύου που χρησιμοποιήθηκαν γι' αυτό το σενάριο προέρχονται από δημοσία σημεία πρόσβασης WiFi μέσα στον χώρο/ στα κτήρια του πανεπιστημίου του Dartmouth κατά τα έτη 2003-2004 [23]. Ο καιρός εκείνου του χρονικού διαστήματος για την περιοχή όπου στεγάζεται το πανεπιστήμιο (Hanover, New Hampshire) συλλέχθηκε από το [24]. Τα δεδομένα συνδυάστηκαν σε ένα κοινό αρχείο και τροφοδότησαν τον SOM, ο οποίος δημιούργησε και οργάνωσε την γνώση για το φορτίο του δικτύου σε χάρτες όμοιους με αυτούς της Εικόνα 9.7.

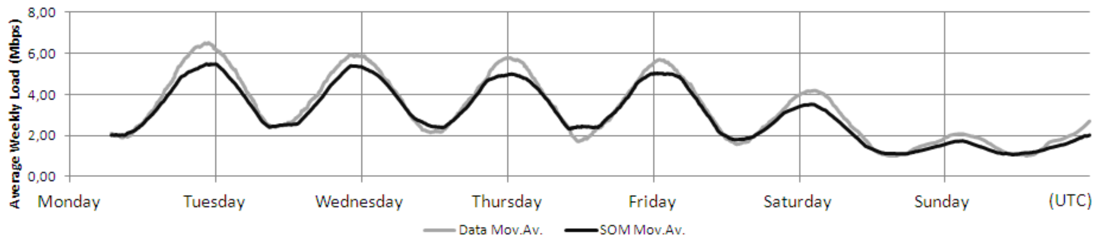
Αυτή η γνώση αξιοποιήθηκε εν συνεχεία από τον μηχανισμό για την πρόβλεψη του φορτίου που θα παρατηρηθεί στο μέλλον στο δίκτυο. Για την αξιολόγηση του μηχανισμού, μέρος των δεδομένων χρησιμοποιήθηκαν για να δημιουργηθούν ερωτήματα προς τον μηχανισμό. Τα αποτελέσματα, οι προβλέψεις δηλαδή που προήλθαν από τον μηχανισμό συγκρίθηκαν διαγραμματικά με τις πραγματικές μετρήσεις του φορτίου κάτω από τις εκάστοτε συνθήκες. Τα αντίστοιχα διαγράμματα προβάλλονται στην Εικόνα 9.8.

Όπως φαίνεται από τα διαγράμματα, ο προτεινόμενος μηχανισμός είναι σε θέση να μάθει σε μεγάλο βαθμό το μοντέλο σύμφωνα με το οποίο αυξομειώνεται το φορτίο του κάθε σημείου πρόσβασης. Συγκεκριμένα, η απόκλιση που παρατηρείται μεταξύ προβλέψεων και πραγματικών τιμών είναι περίπου της τάξης των 2 Mbps.

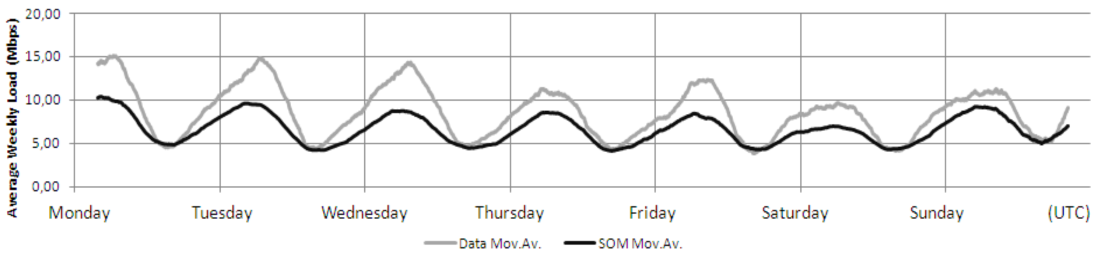


Εικόνα 9.7: Χάρτες SOM για την απεικόνιση του φορτίου ενός δικτύου σύμφωνα με την ώρα, την ημέρα, την περιοχή και τον καιρό στην περιοχή.

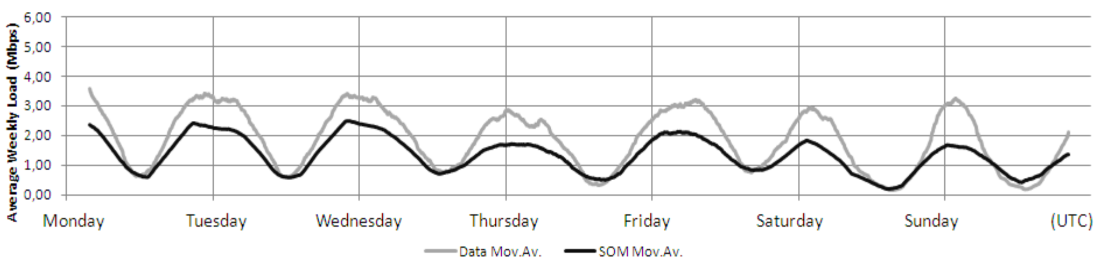
Ο μηχανισμός αποδείχτηκε εξίσου άξιος και για την αξιοποίηση και διαχείριση των δεδομένων σε επίπεδο Big Data. Συγκεκριμένα, το γεγονός ότι ο μηχανισμός στηρίζεται στην τεχνική των SOMs, του επιτρέπει να απεικονίζει πολυδιάστατα δεδομένα στις 2 διαστάσεις του χάρτη της Εικόνα 9.7. Με αυτόν τον τρόπο, δεδομένα από διαφορετικές πηγές μπορούν να οργανωθούν σε πολυδιάστατα δεδομένα και να διαχειρισθούν χωρίς να επιβαρύνουν τον μηχανισμό, ενώ η τελική 2-διάστατη απεικόνισή τους διευκολύνει την ανάλυσή τους από τον χρήστη του μηχανισμού και μειώνει τις αποθηκευτικές απαιτήσεις του συστήματος. Επιπλέον, καθώς τα δεδομένα ομαδοποιούνται σύμφωνα με την ομοιότητά τους, η διατήρηση και αποθήκευση μόνο ενός αντιπροσωπευτικού δείγματος ανά ομάδα είναι αρκετή για την διαφύλαξη της απαραίτητης πληροφορίας. Ως εκ τούτου, μειώνονται περαιτέρω οι απαιτήσεις αποθήκευσης αυτών. Σε αυτό το σενάριο, τα αρχικώς 3.436.771 μη δομημένα δεδομένα, οργανώθηκαν σε 202.163 δεδομένα 17-διαστάσεων που απεικονίστηκαν σε 202.163 διανύσματα πάνω στον δισδιάστατο χάρτη. Τελικά, το πολύ 2.650 διανύσματα ήταν αρκετό να αποθηκευτούν ώστε να διατηρηθεί όλη η χρήσιμη πληροφορία.



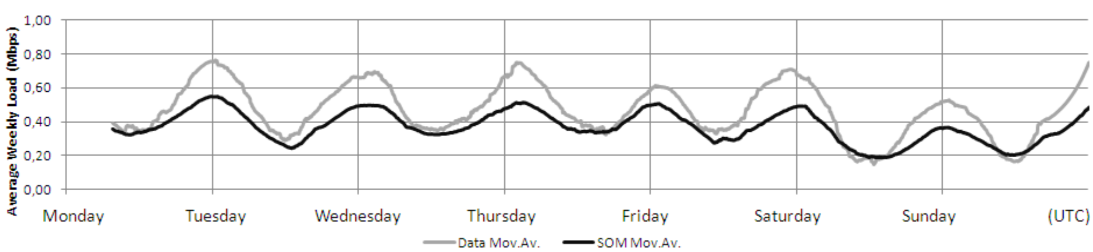
(α) Ακαδημαϊκά κτήρια



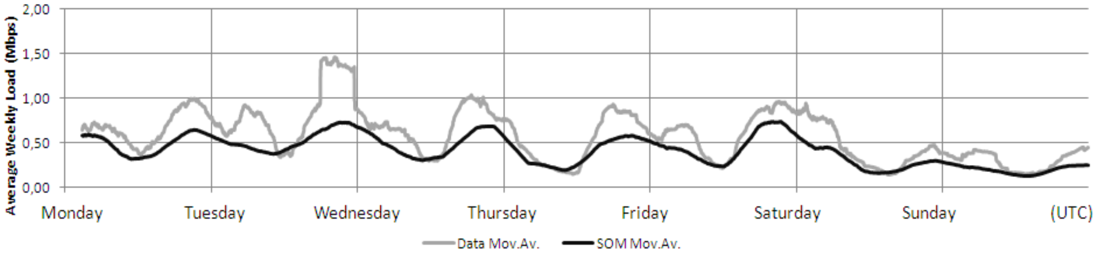
(β) Κατοικίες



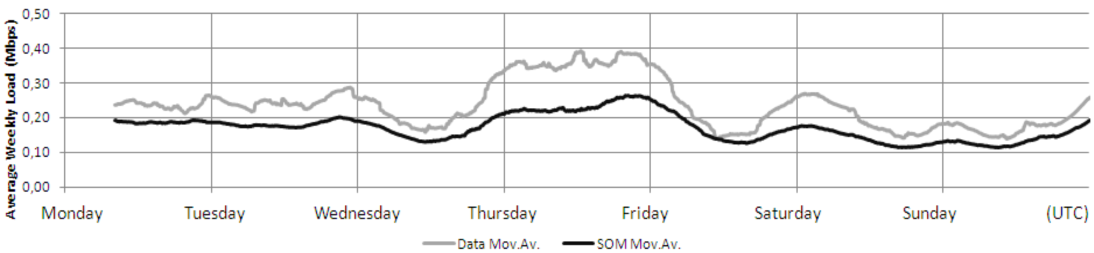
(γ) Βιβλιοθήκες



(δ) Καφετέριες



(ε) Κτήρια διοίκησης

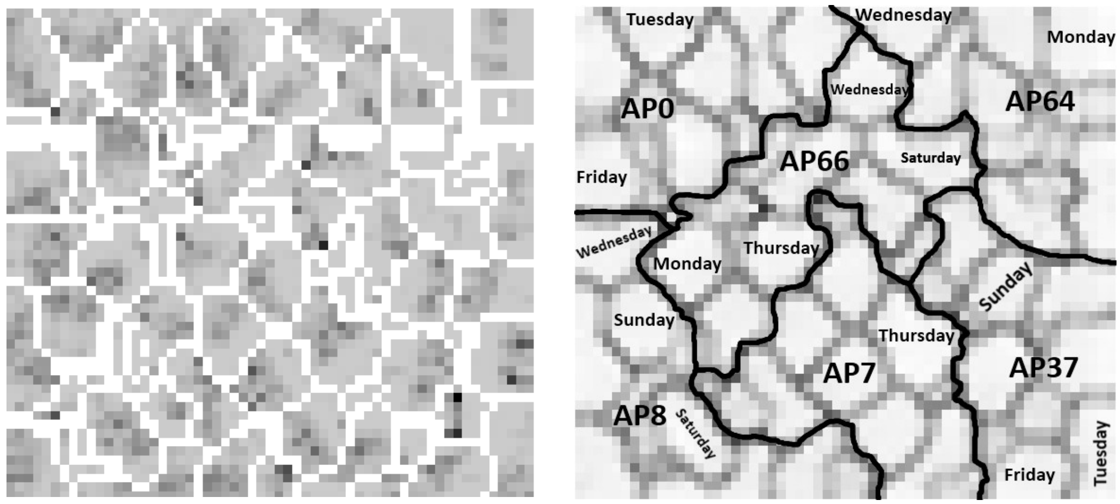


(στ) Αθλητικά κτήρια/ χώροι

Εικόνα 9.8: Συγκριτικά διαγράμματα μεταξύ προβλέψεων και μετρήσεων για κάθε τύπο/ είδος κτηρίου.

9.5.2.2 Μελέτη του φορτίου ενός δικτύου σύμφωνα με την ώρα, την ημέρα, την περιοχή, την εβδομάδα του χρόνου και το αν η εν λόγω μέρα είναι αργία ή όχι

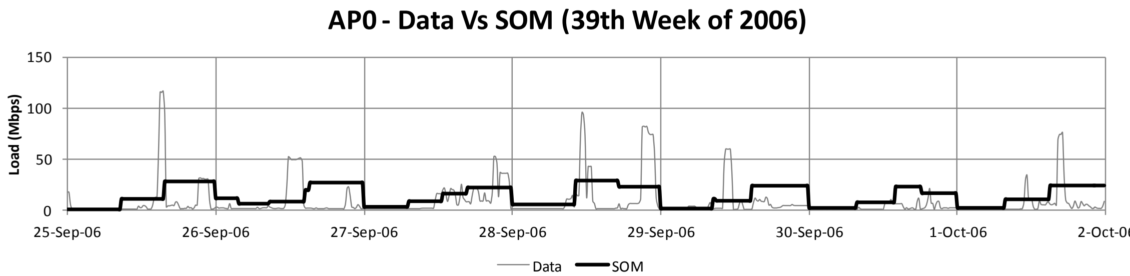
Χρησιμοποιώντας τον ίδιο μηχανισμό, σε αυτό το σενάριο εξετάστηκε η περίπτωση χρήσης της πληροφορίας αν η μέρα που μας ενδιαφέρει είναι κάποια αργία ή όχι και σε ποια εβδομάδα του χρόνου απευθυνόμαστε. Γι' αυτό το σενάριο τα δεδομένα που χρησιμοποιήθηκαν προέρχονται από το δίκτυο ασύρματων σημείων πρόσβασης "Île sans fil" [25] του Montréal, Québec, Canada [26]. Δεδομένα για τις αργίες της περιοχής αντλήθηκαν από ψηφιακά ημερολόγια του διαδικτύου. Οι αντίστοιχοι χάρτες SOM που δημιουργήθηκαν είναι της μορφής της Εικόνα 9.9.

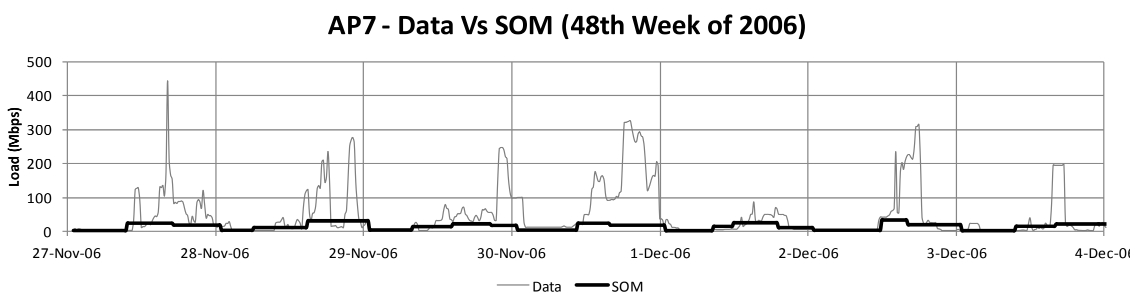


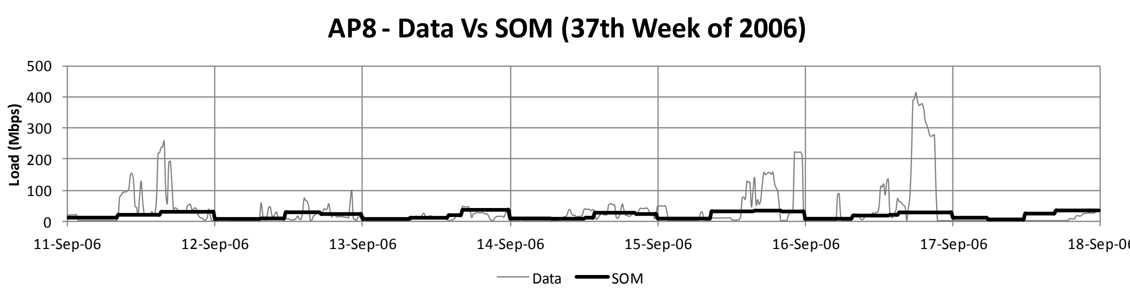
Εικόνα 9.9: Χάρτες SOM για την μελέτη του φορτίου ενός δικτύου σύμφωνα με την ώρα, την ημέρα, την περιοχή, την εβδομάδα του χρόνου και το αν η εν λόγω μέρα είναι αργία ή όχι.

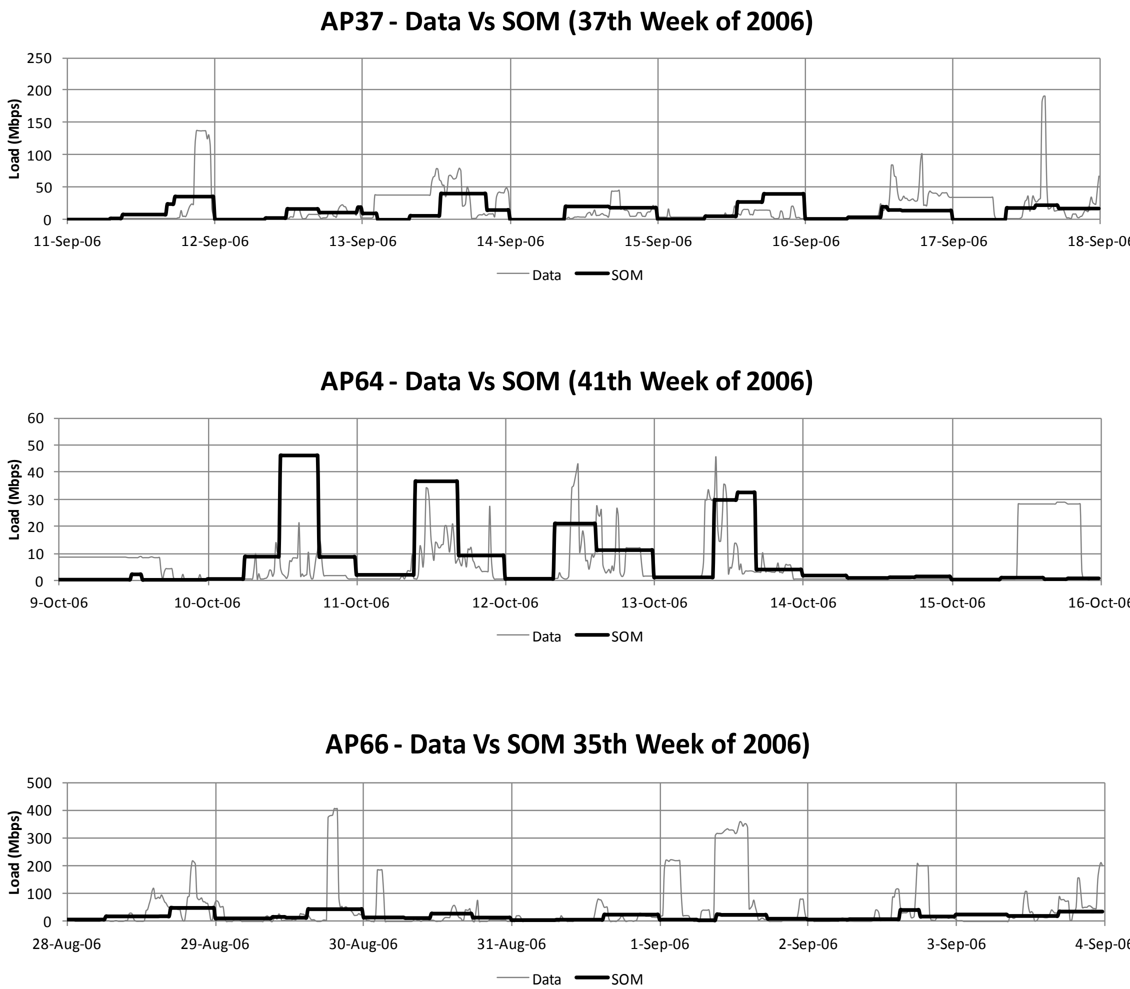
Ενδεικτικά αποτελέσματα από τα συγκριτικά διαγράμματα των εκτιμήσεων του μηχανισμού και των πραγματικών τιμών του φορτίου απεικονίζονται στην Εικόνα 9.10.

Όπως φαίνεται και από τα διαγράμματα, αν και ο μηχανισμός έχει μάθει την τάση του φορτίου του δικτύου, τα αποτελέσματα δεν φαίνονται τόσο ικανοποιητικά όσο αυτά του προηγούμενου σεναρίου. Για την ακρίβεια, η μέση απόκλιση μεταξύ των προβλέψεων και των πραγματικών τιμών είναι της τάξης των 25-26 Mbps. Από την άλλη πλευρά όμως, αυτή η διαφοροποίηση δεν οφείλεται μόνο στον μηχανισμό αυτό κάθε αυτό. Αφενός, η εισαγωγή της παραμέτρου «εβδομάδα του έτους», σηματοδοτεί ότι πλέον παρατηρήσεις της ίδιας ημέρας (π.χ. Δευτέρας) αλλά διαφορετικών εβδομάδων αντιμετωπίζονται εντελώς διαφορετικά σε σχέση με το προηγούμενο σενάριο όπου οι παρατηρήσεις από όλες τις Δευτέρες αντικατόπτριζαν την κίνηση οποιασδήποτε Δευτέρας και άρα μπορούσαν να χρησιμοποιηθούν αθροιστικά. Με άλλα λόγια, τα δεδομένα που έχουν χρησιμοποιηθεί για την εκπαίδευση του μηχανισμού ως προς την κίνηση του δικτύου ανά εβδομάδα έχουν περιοριστεί κατά πολύ (σε 2), ελλείψει παραπάνω δεδομένων. Αφετέρου, η διαφοροποίηση αυτή οφείλεται και σε αστάθμητους παράγοντες που μπορεί να έχουν αυξήσει, συνολικά μέσα στο έτος, κατά πολύ την κίνηση του δικτύου, π.χ., κάποια διαφήμιση που έχει αυξήσει την προσέλευση του κόσμου στην συγκεκριμένη περιοχή ή κάποια βελτίωση στην σχέση τιμής – υπηρεσίας που παρέχεται από το δίκτυο "Île sans fil". Πράγματι, όπως φαίνεται και από τα ενδεικτικά διαγράμματα της Εικόνα 9.11, παρατηρώντας την κίνηση του ίδιου σημείου πρόσβασης (AP0) για το ίδιο χρονικό διάστημα (24η εβδομάδα του έτους), η κίνηση του δικτύου φαίνεται να αυξάνεται από έτος σε έτος, σε όλο το μήκος του χρονικού διαστήματος.









Εικόνα 9.10: Ενδεικτικά συγκριτικά διαγράμματα μεταξύ προβλέψεων και μετρήσεων για κάθε ασύρματο σημείο πρόσβασης (AP).

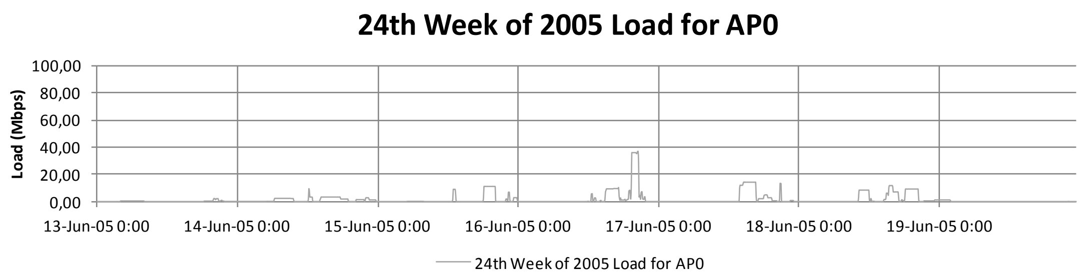
Τα αποτελέσματα του σεναρίου στο θέμα της διαχείρισης και αξιοποίησης των Big Data είναι και σε αυτήν την περίπτωση πολλά υποσχόμενα. Συγκεκριμένα, σε αυτό το σενάριο κάθε έτος περιγραφόταν από 3.574.080 μη δομημένα δεδομένα τα οποία αξιοποιήθηκαν ως 210.240 δομημένα δεδομένα των 17 διαστάσεων ανά έτος. Τελικά, τα δεδομένα αυτά απεικονίστηκαν ως 210.240 δισδιάστατα διανύσματα πάνω στον χάρτη της Εικόνα 9.9 κατά την εκπαίδευση του μηχανισμού, ενώ μόλις 900 από αυτά χρειάστηκε να αποθηκευτούν για την διατήρηση της πληροφορίας που έφεραν.

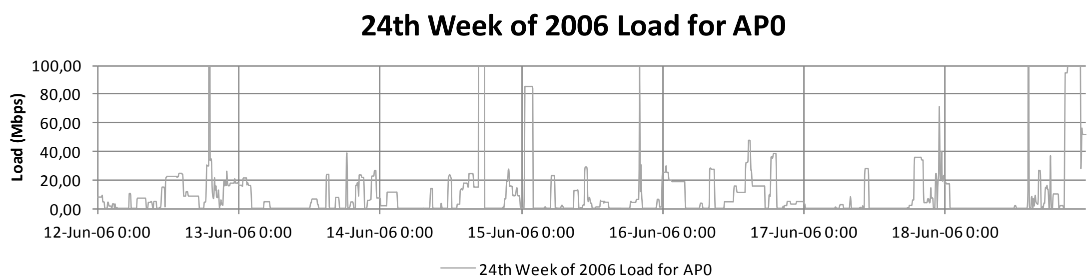
9.5.2.3 Μελέτη του φορτίου ενός δικτύου σύμφωνα με όλες τις παραπάνω παραμέτρους

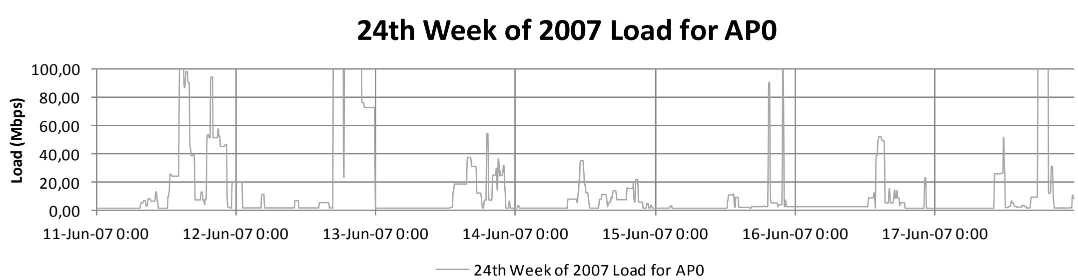
Το 3ο και τελευταίο σενάριο επιχειρεί τον συνδυασμό της πληροφορίας όλων των παραπάνω πηγών στοχεύοντας στην πιο ακριβή πρόβλεψη της κίνησης του δικτύου. Και σε αυτό το σενάριο, τα δεδομένα που χρησιμοποιήθηκαν είναι του δικτύου ασύρματων σημείων πρόσβασης "Île sans fil" [25] του Montréal, Québec, Canada [26] ενώ οι αντίστοιχες πληροφορίες για τον καιρό της περιοχής (θερμοκρασία και βροχόπτωση) ανακτήθηκαν από [27] χρησιμοποιώντας σαν σημείο αναφοράς το Διεθνές αεροδρόμιο του Montréal – Pierre Elliott Trudeau. Οι αντίστοιχοι χάρτες SOM και τα συγκριτικά διαγράμματα των προβλέψεων και των μετρήσεων φαίνονται στην Εικόνα 9.12 και Εικόνα 9.13.

Και σε αυτήν την περίπτωση τα συμπεράσματα είναι όμοια με αυτά του κεφαλαίου 9.5.2.2, δηλαδή: ο μηχανισμός έχει μάθει την τάση της κίνησης του δικτύου και την συμπεριφορά του αλλά υπάρχουν αρκετά μεγάλες αποκλίσεις της τάξης των 26 Mbps κατά μέσο όρο. Από την άλλη, το γεγονός ότι το συμπέρασμα αυτό συμπίπτει με το προηγούμενο, και έχοντας χρησιμοποιήσει το ίδιο πακέτο δεδομένων, ενισχύει την πιθανότητα οι αποκλίσεις να οφείλονται στο πακέτο των δεδομένων και όχι στις επιλεχθείσες παραμέτρους.

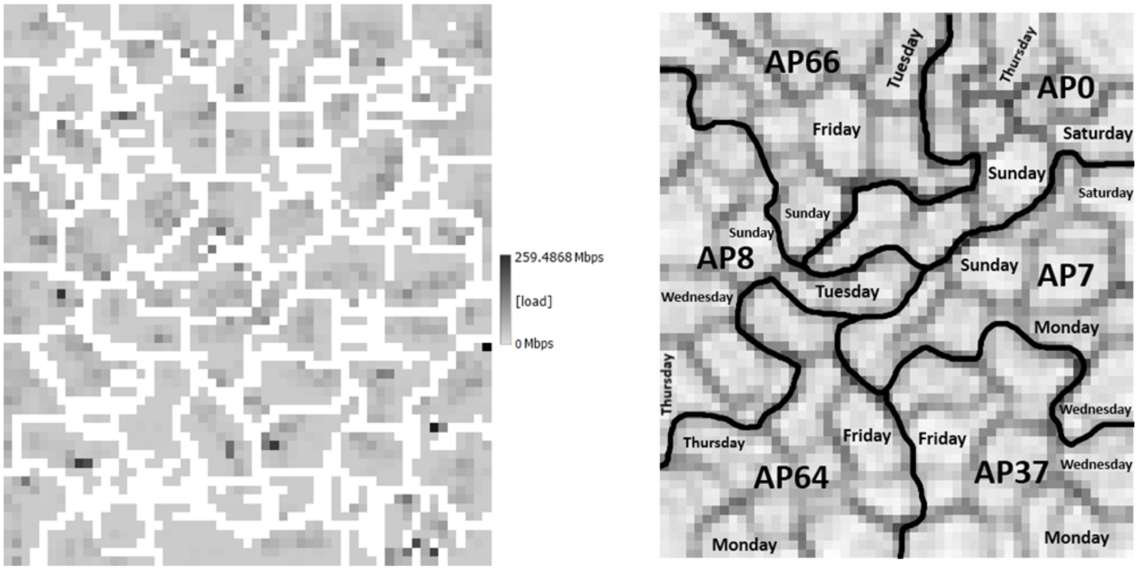
Τέλος, στο γενικό πλαίσιο των θεμάτων που αφορούν την έννοια των Big Data, τα αποτελέσματα για μια ακόμη φορά επιβεβαιώνουν την ικανότητα του μηχανισμού στην διαχείρισή τους και την μείωση των απαιτήσεων αποθηκευτικού χώρου καθώς ο μηχανισμός διαχειρίστηκε και συνδύασε την πληροφορία 3.994.560 μη δομημένων δεδομένα ανά έτος ως 210.240 δεδομένα 19 διαστάσεων. Αυτά απεικονίστηκαν στον χάρτη της Εικόνα 9.12 σαν 210.240 δισδιάστατα δεδομένα και εν τέλει χρειάστηκε να αποθηκευτούν μόλις 928 διανύσματα για την διατήρηση της πληροφορίας.





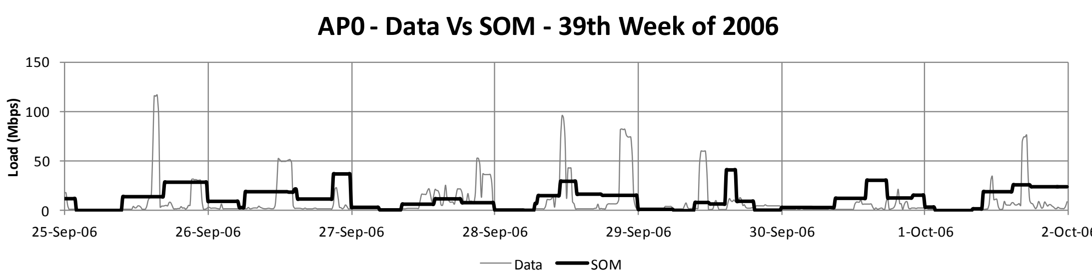


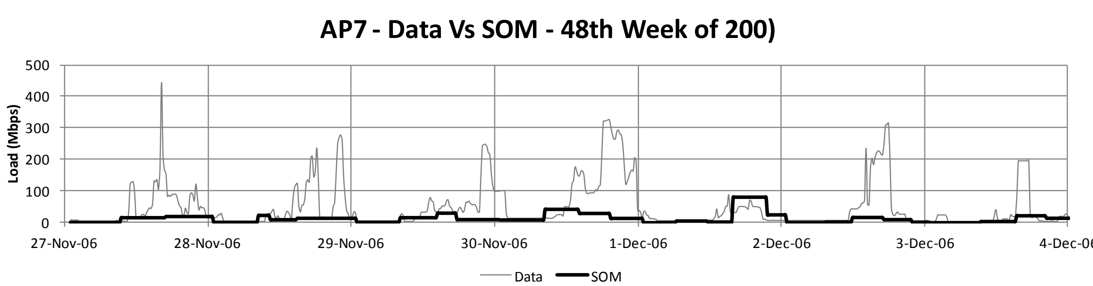
Εικόνα 9.11: Φορτίο κίνησης του δικτύου για συγκεκριμένο σημείο πρόσβασης (AP0) και χρονικό διάστημα για τα έτη 2005, 2006 και 2007.

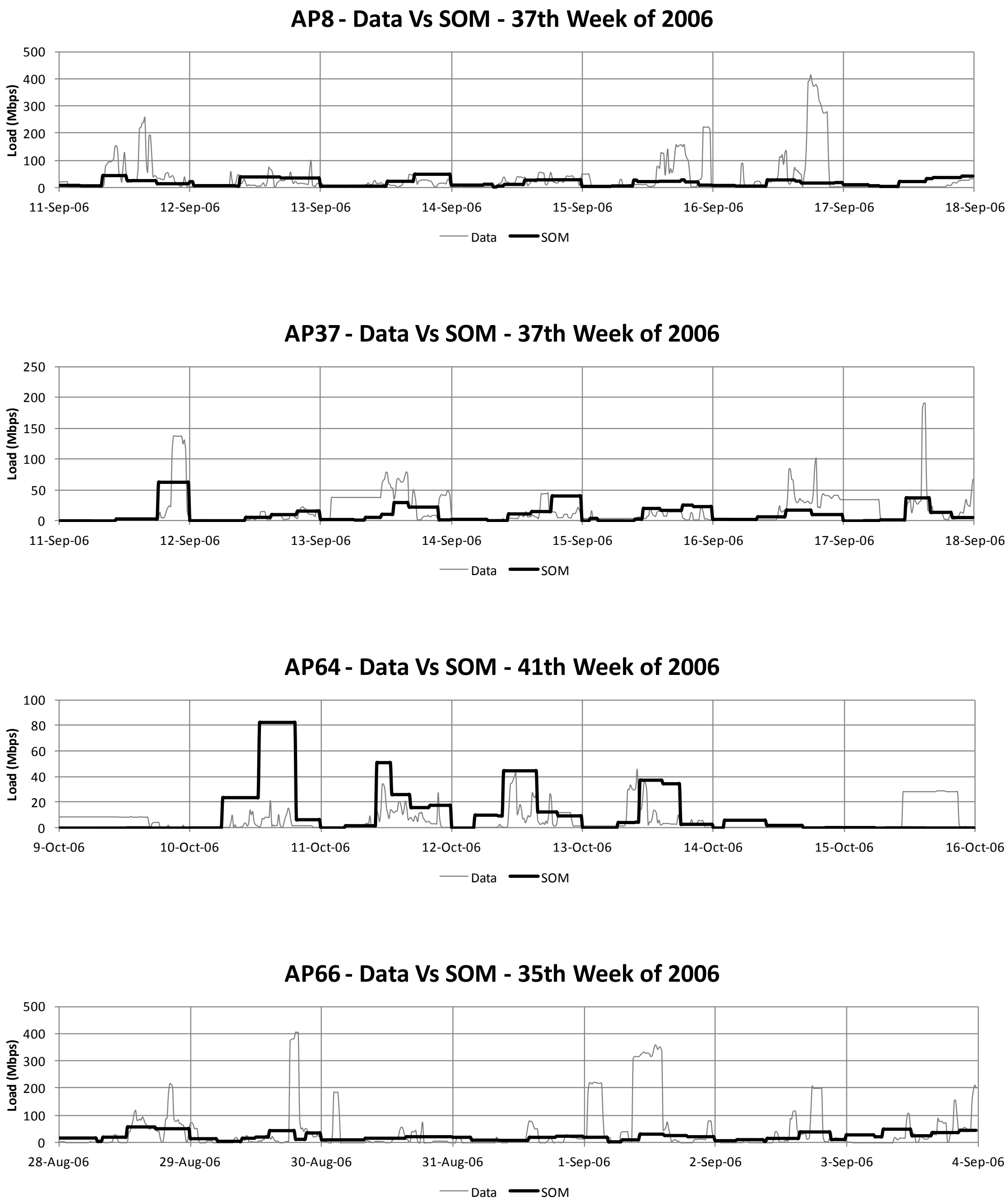


Εικόνα 9.12: Χάρτες SOM για την μελέτη του φορτίου ενός δικτύου σύμφωνα με την ώρα, την ημέρα, την περιοχή, την εβδομάδα του χρόνου, τις αργίες και τον καιρό στην περιοχή.

Αναλυτική παρουσίαση αυτού του μηχανισμού, του σεναρίου και της σύγκρισης των 3ων σεναρίου παρατίθεται στο [28] .







Εικόνα 9.13: Συγκριτικά διαγράμματα των προβλέψεων του μηχανισμού και των αντίστοιχων μετρήσεων για το 3ο σενάριο.

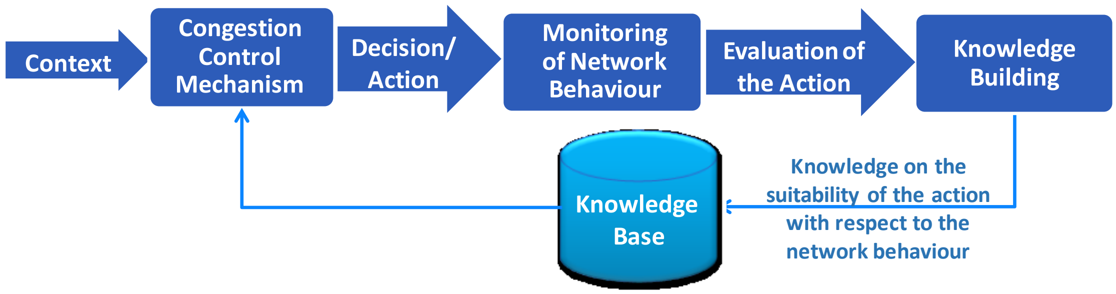
9.6 Γνωσιακή Ανάλυση Δεδομένων Επιπέδου Μεταφοράς

Για την αποφυγή της συμφόρησης στο δίκτυο υπάρχουν και χρησιμοποιούνται διάφοροι αλγόριθμοι με τους πιο διαδεδομένους να είναι ο TCP Reno [29], ο New Reno [30], ο Cubic [31], ο Tahoe [32] και ο TCP Vegas [33]. Από αυτούς, ο μοναδικός αλγόριθμος που λειτουργεί προληπτικά, πριν δηλαδή χάσει κάποια πακέτα, είναι ο TCP Vegas. Ωστόσο, λόγω της αδυναμίας του να προσαρμόζεται στα δυναμικά περιβάλλοντα πολλές φορές μπορεί να παρερμηνεύσει την συμπεριφορά του δικτύου ως μια επερχόμενη συμφόρηση και να μειώσει την ταχύτητα μετάδοσης των δεδομένων χωρίς αυτό να είναι απαραίτητο.

Τέτοιες περιπτώσεις για παράδειγμα αποτελούν η δρομολόγηση των δεδομένων μέσα από μια διαφορετική διαδρομή, υψηλότερου κόστους σε όρους χρόνου αμφίδρομης μεταφοράς (RTT). Αυτό συμβαίνει διότι ο εν λόγω αλγόριθμος λαμβάνει προληπτικά μέτρα για την αποφυγή συμφόρησης βάσει του RTT που παρατηρεί και χωρίς να έχει γνώση αν μια αύξηση αυτού προκύπτει από την αύξηση της κίνησης στο δίκτυο ή από μια διαφορετική δρομολόγηση. Το αποτέλεσμα μιας τέτοιας παρερμηνείας του TCP Vegas, προκαλεί την αποσταθεροποίηση του δικτύου σε όρους χρήσης των δυνατοτήτων του. Για παράδειγμα, συχνές αλλαγές στην δρομολόγηση των δεδομένων προκαλεί συχνές μη απαραίτητες αλλαγές στο παράθυρο συμφόρησης της ζεύξης και στην χρησιμοποίησή της.

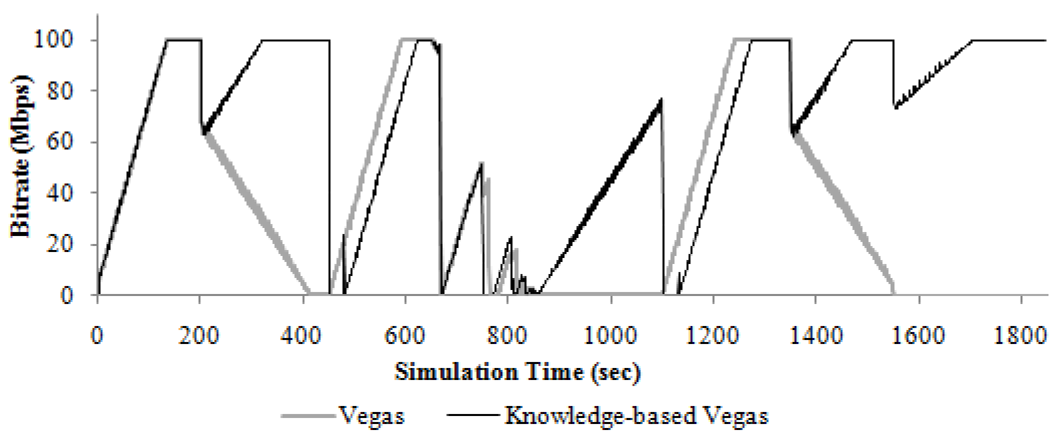
Για τον λόγο αυτό, ο μηχανισμός που περιγράφεται παρακάτω στοχεύει στην παραγωγή γνώσης και εμπειρίας τέτοιας που να μπορεί να καθοδηγεί τον TCP Vegas για το αν πρόκειται για κάποια άλλη αιτία που προκάλεσε την αύξηση του RTT ή αν πράγματι είναι συμφόρηση. Και σε αυτήν την περίπτωση χρησιμοποιήθηκε η τεχνική SOM για την δημιουργία γνώσης. Ο μηχανισμός λειτουργεί σύμφωνα με την Εικόνα 9.14.

Έτσι, αρχικά ο TCP Vegas λαμβάνει την απόφαση να προχωρήσει ή όχι σε αναπροσαρμογή του παράθυρου συμφόρησης. Η απόφαση αυτή μαζί με τις συνθήκες που προκάλεσαν αυτήν την απόφαση καθώς και η αντίδραση/ συμπεριφορά του δικτύου μετά την εφαρμογή της απόφασης στέλνονται στην λειτουργία της μηχανικής μάθησης για να χτίσουν την εμπειρία του μηχανισμού ως προς την ορθότητα αυτής της απόφασης. Η ορθότητα η μη της απόφασης προκύπτει από την απώλεια ή όχι πακέτων. Αξίζει να σημειωθεί εδώ ότι αρχικά, και μέχρι να αποκτήσει την απαραίτητη εμπειρία, ο μηχανισμός συμβουλεύει πάντα τον TCP Vegas να αγνοήσει αυτήν την αλλαγή στο RTT. Τελικά ο μηχανισμός έχει αποκτήσει αρκετή εμπειρία ώστε να συμβουλεύει ορθά τον αλγόριθμο αποφυγής συμφόρησης.



Εικόνα 9.14: Επισκόπηση του προτεινόμενου μηχανισμού.

Ενδεικτικά αποτελέσματα αυτού του μηχανισμού φαίνονται στο διάγραμμα της Εικόνα 9.15 ενώ μια πιο πλήρης ανάλυση του μηχανισμού και της μελέτης γενικότερα μπορεί να βρεθεί στο [34].

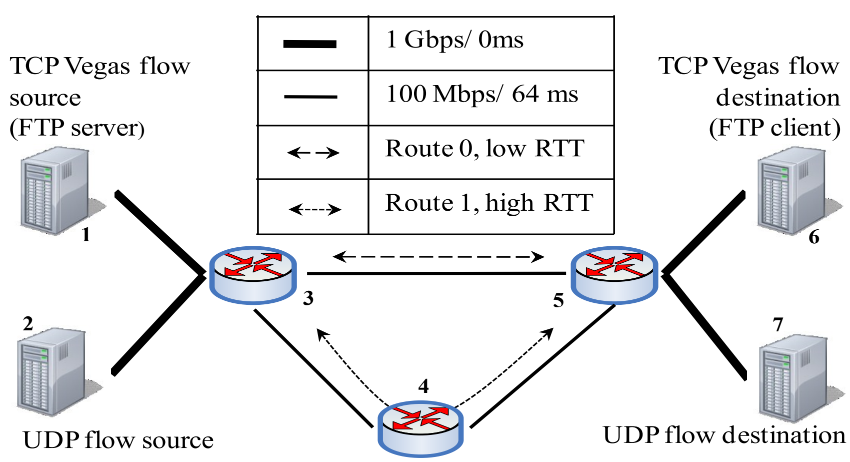


Εικόνα 9.15: Συγκριτικό διάγραμμα του ρυθμού μετάδοσης δεδομένων της γραμμής όταν χρησιμοποιείται μόνο ο αλγόριθμος αποφυγής συμφόρησης (γκρι γραμμή) και όταν ο τελευταίος συνδυάζεται με την γνώση που παράγεται από τον προτεινόμενο μηχανισμό (μαύρη γραμμή).

Το σενάριο που χρησιμοποιήθηκε στην περίπτωση της Εικόνα 9.15 έχει ως εξής: έχουμε την τοπολογία της Εικόνα 9.16, παρακολουθούμε τα πακέτα που στέλνονται από τον κόμβο 1 στον κόμβο 6 με αρχική διαδρομή την 1-3-5-6 και προς τις 2 κατευθύνσεις και τα παρακάτω γεγονότα:

- T = 200sec : αλλαγή δρομολόγησης (στην κατεύθυνση προς τον κόμβο 6) με αύξηση της μικρότερης τιμής RTT που έχει παρατηρηθεί από τον TCP Vegas, νέα διαδρομή 1-3-4-5-6
- T = 450sec: επιστροφή στην διαδρομή 1-3-5-6
- T = 650sec: είσοδος UDP ροής στην ζεύξη 3-5
- T= 850sec: αλλαγή δρομολόγησης λόγω συμφόρησης με αύξηση της μικρότερης τιμής RTT που έχει παρατηρηθεί από τον TCP Vegas, νέα διαδρομή 1-3-4-5-6
- T = 1100sec: επιστροφή στην διαδρομή 1-3-5-6
- T = 1350sec: αλλαγή δρομολόγησης (στην κατεύθυνση από τον κόμβο 6) με αύξηση της μικρότερης τιμής RTT που έχει παρατηρηθεί από τον TCP Vegas, νέα διαδρομή 6-5-4-3-1
- T = 1550sec: επιστροφή στην διαδρομή 6-5-4-3-1

Όπως παρατηρείται από το συγκριτικό διάγραμμα της Εικόνα 9.15, και στις 3 περιπτώσεις αλλαγής δρομολόγησης των πακέτων (T = 200, 850 και 1350 sec) ο μηχανισμός έχει συμβουλεύσει σωστά τον αλγόριθμο αποφυγής συμφόρησης, εξασφαλίζοντας έτσι στο δίκτυο μεγαλύτερες ταχύτητες μεταφοράς δεδομένων και λιγότερες διακυμάνσεις και αστάθειες σε αυτήν.

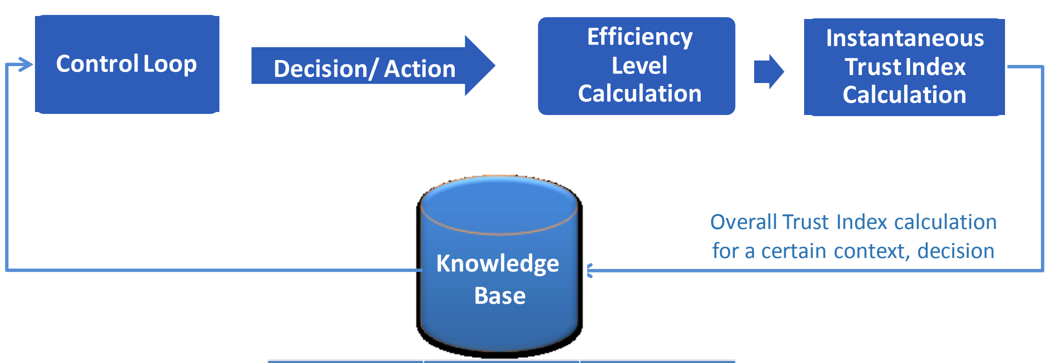


Εικόνα 9.16: Τοπολογία δικτύου προσομοίωσης.

9.7 Μηχανισμοί για την Ανάπτυξη και Διαχείριση Εμπιστοσύνης (Trust Management)

Ο μηχανισμός που περιγράφεται εδώ στοχεύει στο να εμφυσήσει σε ένα βρόχο ελέγχου τη γνώση, αν οι αποφάσεις του είναι αρκετά αξιόπιστες. Η γνώση αυτή μπορεί να αξιοποιηθεί έτσι ώστε να βελτιωθούν οι επιδόσεις του βρόχου ελέγχου σε επίπεδο εμπιστοσύνης μέσω του αποκλεισμού των αποφάσεων που δεν επιτυγχάνουν τα επιθυμητά επίπεδα αξιοπιστίας. Πιο συγκεκριμένα, ο μηχανισμός αποθηκεύει πληροφορίες σχετικά με τις καταστάσεις που αντιμετωπίζει ο βρόχος ελέγχου, συμπεριλαμβανομένης της σχετικής απόφασης που εφαρμόστηκε για τον χειρισμό τους σε μια βάση γνώσης όμοιας με αυτή που απεικονίζεται στην Εικόνα 9.17.

Αυτό επιτρέπει την αξιοποίηση παρελθόντων αλληλεπιδράσεων, ώστε να καταστεί δυνατή η ταχύτερη και αποτελεσματικότερη αντιμετώπιση των προβλημάτων. Έτσι, ένας βρόχος ελέγχου μπορεί να επιλέξει την πιο αξιόπιστη απόφαση δεδομένων των συνθηκών που επικρατούν στο σύστημα. Επιπλέον, η διαδικασία λήψης αποφάσεων μπορεί να βελτιωθεί σε επίπεδο μείωσης του χρόνου που απαιτείται για την επιλογή μιας συγκεκριμένης ενέργειας. Ως πρώην «αξιόπιστη» απόφαση μπορεί να εφαρμοστεί μέσα στις ίδιες συνθήκες του συστήματος, χωρίς την ανάγκη της εκτέλεσης μια διαδικασία βελτιστοποίησης.



State ID	Decision ID	Trust Index
201201001	201201101	60%
201201002	201201201	75%
201202003	201201202	52%
201201001	201201202	80%

Εικόνα 9.17: Επισκόπηση του προτεινόμενου μηχανισμού.

Έτσι, κάθε φορά που λαμβάνεται μια απόφαση, ο βρόχος ελέγχου παρέχει πληροφορίες σχετικά με την επικρατούσα κατάσταση (παραμέτρους που ενεργοποίησαν το βρόχο ελέγχου) και την αντίστοιχη απόφαση που έλαβε. Το επίπεδο αποδοτικότητας της απόφασης υπολογίζεται, λαμβάνοντας υπόψη διαφορετικά κατά περίσταση (ανάλογα με τον βρόχο ελέγχου) κριτήρια, από την σχέση (9.1):

$$EL(t) = F(m_1, m_2, ..., m_n) \tag{9.1}$$

όπου F μπορεί να είναι οποιαδήποτε συνάρτηση, όπως άθροισμα με βάρη (weighted sum), που συνδέει τα κριτήρια m_1, m_2, …., m_n. Παραδείγματα των κριτηρίων μπορεί να είναι α) η απόκλιση από τον επιθυμητό στόχο του βρόγχου ελέγχου (π.χ., επίπεδα ποιότητας υπηρεσίας), β) πόροι που χρειάστηκαν για την πραγματοποίηση των αποφάσεων του βρόχου, γ) χρόνος που χρειάστηκε για την πραγματοποίηση των αποφάσεων του βρόχου, δ) αριθμός αναδιαμορφώσεων (reconfigurations) που προκλήθηκαν λόγω της απόφασης/ ενέργειας του βρόγχου κλπ.

Στην συνέχεια, όπως φαίνεται και από την Εικόνα 9.17 υπολογίζονται ο στιγμιαίος δείκτης εμπιστοσύνης (instantaneous trust index) και ο συνολικός δείκτης (overall trust index). Για του υπολογισμούς αυτούς ακολουθείται το μοντέλο της τεχνικής Q-learning, και ως εκ τούτου οι δείκτες δίνονται από τις σχέσεις (9.2) - (9.3) και (9.4) - (9.5) αντίστοιχα.

$$EL(t) \begin{cases} < EL_{thres}, & r(s(t), a(t)) < 0 \\ > EL_{thres}, & r(s(t), a(t)) > 0 \end{cases} \tag{9.2}$$

$$EL_{thres} = F(m_{1,opt}, m_{2,opt}, ..., m_{n,opt}) \tag{9.3}$$

$$Q(s(t), a(t)) = \left\langle \sum_{t=0}^{\infty} \gamma^t r(s(t), a(t)) \right\rangle_{s,r} \tag{9.4}$$

$$Q(s(t), a(t)) \to Q(s(t), a(t)) + \varepsilon[r(t) + \gamma \max_{a(t+1)} Q(s(t+1), a(t+1)) - Q(s(t), a(t))] \tag{9.5}$$

όπου r(s(t),a(t)) είναι η ανταμοιβή για την απόφαση του κόμβου, s(t) είναι η κατάσταση στην οποία βρίσκεται ο βρόχος, a(t) είναι η απόφαση του βρόχου, $m_{i,opt}$ είναι η ελάχιστη επιθυμητή τιμή του κριτηρίου mi, το $\langle \ \rangle_{s,r}$ αναπαριστά την μέση τιμή, το γ εκφράζει το πόσο επηρεάζουν οι παλιότερες παρατηρήσεις τον υπολογισμό του ολικού δείκτη εμπιστοσύνης και το ε το ρυθμό προσαρμογής του ολικού δείκτη εμπιστοσύνης στις παρατηρήσεις.

Περισσότερες λεπτομέρειες ως προς την μοντελοποίηση του μηχανισμού μπορούν να βρεθούν και στο [35].

9.8 Βιβλιογραφία Κεφαλαίου

[1] K. Tsagkaris, A. Katidiotis, P. Demestichas, "Neural Network-based Learning schemes for Cognitive Radio systems", Computer Communications, Vol. 31, Issue 14, pp. 3394-3404, 2008

[2] A. Katidiotis, K. Tsagkaris, P. Demestichas, "Performance Evaluation of Artificial Neural Network-Based Learning Schemes for Cognitive Radio Systems", Computers & Electrical Engineering, Volume 36, Issue 3, Pages 518-535, May 2010

[3] N. Baldo, B. R. Tamma, B. S. Manoj, R. Rao and M. Zorzi, "A neural network based cognitive controller for dynamic channel selection", In Proceedings of the 2009 IEEE international Conference on Communications, Dresden, Germany, June 14 - 18, 200

[4] A. M. Balieiro, K. L. Dias, A. R. G. Castro, E. R. Alves, "Handoff de Espectro em Redes Baseadas em Rádio Cognitivo Utilizando Redes Neurais Artificiais", XXX Congreso Da Sociedade Brasileira De Computacao (CSBC 2010), in portuguese, online available at: http://www.inf.pucminas.br/sbc2010/anais/pdf/wperformance/st03_03.pdf

[5] O. Orcay, B. Ustundag, "Pattern recognition in cognitive communication", Computer and Information Sciences, 2008, ISCIS '08, 23rd International Symposium on, vol., no., pp.1-6, 27-29, Oct. 2008

[6] P. Demestichas, A. Katidiotis, K. Tsagkaris, E. Adamopoulou, K. Demestichas, "Enhancing Channel Estimation in Cognitive Radio Systems by means of Bayesian Networks", Wireless Personal Communications, vol 49, Issue 1, pp 87-105, 2009

[7] V. Stavroulaki, Y. Kritikou, P. Demestichas, "Acquiring and learning user information in the context of cognitive device management", In Proc. IEEE International Conference on Communications 2009 (ICC 2009), Dresden Germany, June 2009

[8] Y. Kritikou, V. Stavroulaki, P. Demestichas, "Learning user preferences for the realization of intuitive cognitive devices", submitted for publication to the Wireless Personal Communications

[9] N. Baldo, M. Zorzi, "Cognitive Network Access using Fuzzy Decision Making", Communications, 2007. ICC '07. IEEE International Conference on , vol., no., pp.6504-6510, 24-28 June 2007

[10] N. Baldo, M. Zorzi, "Fuzzy Logic for Cross-layer Optimization in Cognitive Radio Networks", 1st IEEE Workshop on Cognitive Radio Networks, (in conjunction with IEEE CCNC 2007), Las Vegas, Nevada, USA, January 2007

[11] L. Bixio, M. Ottonello, H. Sallam, M. Raffetto, C. S. Regazzoni, "Signal Classification based on Spectral Redundancy and Neural Network Ensembles", in Proc. IEEE CROWNCOM 2009, Hannover, Germany, 22nd – 24th June 2009

[12] European Telecommunications Standards Institute (ETSI) Group Specification (GS) for Autonomic network engineering for the self-managing Future Internet (AFI), "Autonomic network engineering for the self-managing Future Internet (AFI); Generic Autonomic Network Architecture (An Architectural Reference Model for Autonomic Networking, Cognitive Networking and Self-Management)", ETSI GS AFI 002 V1.1.1 (2013-04)

[13] A. Bantouna, K. Tsagkaris, P. Demestichas, "Knowledge Functional Block for E-UTRAN", Accepted to 5th IEEE International Workshop on Management of Emerging Networks and Services (IEEE MENS 2013), Atlanta , GA USA, Dec. 2013

[14] 3GPP Technical Specification, "LTE; Telecommunication management; Performance Management (PM); Performance measurements Evolved Universal Terrestrial Radio Access Network (E-UTRAN)", 3GPP TS 32.425, v. 11.4.0, Release 11

[15] O. Holland, H. Aghvami, S. Romaszko, D. Denkovski, V. Pavlovska, L. Gavrilovska, A. Bantouna, V. Stavroulaki, Y. Kritikou, P. Demestichas, E. Jorswieck, "The ICT-ACROPOLIS Network of Excellence: Spectrum Sharing and Coexistence based on the Cognition Cycle", Future Network & Mobile Summit 2013 (FUNEMS 2013), Lisbon, Portugal, July 3-4, 2013

[16] V. Stavroulaki, A. Bantouna, Y. Kritikou, K. Tsagkaris, P. Demestichas, P. Blasco, F. Bader, M. Dohler, D. Denkovski, V. Atanasovski, L. Gavrilovska, K. Moessner, "Knowledge Management Toolbox: Machine Learning for Cognitive Radio Network", Published in IEEE Vehicular Technology Magazine, Special Issue on

"Applications of Cognitive Radio Networks", vol.7, no.2, pp.91-99, June 2012, doi: 10.1109/MVT.2012.2190196, URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6184367&isnumber=6201347

[17] T. Kohonen, Self-Organizing Maps, Series in Information Sciences, 2nd ed., vol. 30, Springer, Heidelberg, 1997

[18] T. Kohonen, The Self-Organizing map, Elsevier, Neurocomputing 21 1-6 , 1998

[19] K. Tsagkaris, A. Bantouna, P. Demestichas, Self-Organizing Maps for Advanced Learning in Cognitive Radio Systems, Accepted for Publication in Computer and Electrical Engineering, Elsevier, March 2012, http://dx.doi.org/10.1016/j.compeleceng.2012.03.008

[20] A. Bantouna, V. Stavroulaki, Y. Kritikou, K. Tsagkaris, P. Demestichas, K. Moessner, "An overview of learning mechanisms for cognitive systems", Published to EURASIP Special Issue on Ten Years of Cognitive Radio: State of the Art and Perspectives, EURASIP Journal on Wireless Communications and Networking 2012, 2012:22 doi:10.1186/1687-1499-2012-22, January 2012

[21] A. Bantouna, K. Tsagkaris, V. Stavroulaki, P. Demestichas, "Learning Techniques for Context Diagnosis and Prediction in Cognitive Communications", to be published in Cognitive Communications: Distributed Artificial Intelligence (DAI), Regulatory Policy & Economics, Implementation. H. Zhang and D. Grace, J. Wiley and Sons, October 2012.

[22] M. Ghader, A. Bantouna, L. Bennacer, G. Calochira, B. Fuentes, G. Katsikas, Z. Yousaf, "On Accomplishing Context Awareness for autonomic network management", accepted at Future Network and Mobile Summit 2012, 4 - 6 July 2012, Berlin, Germany

[23] Trace: dartmouth/campus/snmp/fall0304 (v. 2004-11-09), available at http://crawdad.cs.dartmouth.edu/meta.php?name=dartmouth/campus#N1019E, last accessed: July 23rd, 2012

[24] The Old Farmer's Almanac website, weather history, http://www.almanac.com/weather/history, last accessed: July 23rd, 2012

[25] The Île Sans Fil website: http://www.ilesansfil.org/, last accessed: August 10th, 2012

[26] Trace ilesansfil/wifidog/session/04_07 (v. 2007-08-27), available at http://crawdad.cs.dartmouth.edu/meta.php?name=ilesansfil/wifidog#N1006B, last accessed August 10th, 2012

[27] Website, http://www.wunderground.com/history/, last accessed August 15th, 2012

[28] A. Bantouna, G. Poulios, K. Tsagkaris, P. Demestichas, "Network Load Predictions based on Big Data and the Utilization of Self-Organizing Maps", JNSM Special Issue 2013 : Springer Journal of Network and Systems Management – Special Issue on "Data Mining for Monitoring and Managing Systems and Networks", Sept. 2013, DOI: 10.1007/s10922-013-9285-1, Volume 22, Issue 2 (2014), Page 150-173

[29] K. Fall and S. Floyd, "Simulation-based comparisons of Tahoe, Reno, and Sack TCP", ACM Comput. Commun. Rev., Vol. 26, No. 3, pp.5 - 21 , 1996

[30] S. Floyd, T. Henderson, and A. Gurtov, "The NewReno Modification to TCP's Fast Recovery Algorithm", RFC3782, April 2004

[31] Sangtae Ha, Injong Rhee, Lisong Xu, "CUBIC: a new TCP-friendly high-speed TCP variant", SIGOPS Oper. Syst. Rev., Vol. 42, No. 5. (July 2008), pp. 64-74. doi:10.1145/1400097.1400105

[32] Jacobson, V.; , "Congestion avoidance and control," ACM SIGCOMM Computer Communication Review , vol.18, no.4, pp.314-329, August 1988 doi: 10.1145/52325.52356

[33] Brakmo, L.S.; Peterson, L.L.;, "TCP Vegas: end to end congestion avoidance on a global Internet," Selected Areas in Communications, IEEE Journal on , vol.13, no.8, pp.1465-1480, Oct 1995. doi: 10.1109/49.464716

[34] A. Bantouna, K. Tsagkaris, G. Poulios, A. Manzalini, P. Demestichas, "Knowledge in Support of Congestion Control Mechanisms", in proceedings of Future Network Mobile Summit (FuNeMS) 2012, 4 - 6 July 2012, Berlin, Germany

[35] L. Ciavaglia, S. Ghamri - Doudane, M. Smirnov, P. Demestichas, V. Stavroulaki, A. Bantouna, Unifying Management of Future Networks with Trust, accepted for publication in Bell Labs Technical Journal (BLTJ), Special Issue On "Delivering Network Assurance through Secure and Reliable Products, Software, Services and Solutions, December 2011

10 ACRONYMS

Acronym	Explanation
1 – 9	
3GPP	Third Generation Partnership Project
A	
AFI	Autonomic network engineering for the self-managing Future Internet
AP	Access Point
B	
BI	Business Intelligence
BMU	Best Matching Unit
C	
CAPEX	Capital Expenditures
CPT	Conditional Probability Table
CRSs	Cognitive Radio Systems
D	
DE	Decision-making-Elements
DeNB	Donor eNodeB
E	
EL	Efficiency Level
eNB	eNodeB
ETSI	European Telecommunications Standards Institute
E-UTRAN	Evolved Universal Terrestrial Radio Access Network

E-RAB	E-UTRAN Radio Access Bearer
F	
FTP	File Transfer Protocol
G	
GANA	Generic Autonomic Network Architecture
GUI	Graphical User Interface
H	
I	
IoT	Internet of Things
ITI	Instantaneous Trust Index
J	
K	
L	
M	
MAE	Mean Absolute Error
MBTS	Model-Based-Translation Service
MIB	Management Information Base
MSE	Mean Square Error
N	
NN	Neural Network
NOs	Network Operators
NS-2	Network Simulator version 2
O	
ONIX	Overlay Network for Information eXchange

OPEX	Operational Expenditures
OTI	Overall Trust Index
P	
P2P	Peer-to-Peer
PDF	Probability Density Function
PLGSOM	Parameterless-Growing Self-Organizing Map
Q	
QCI	Quality Channel Indicator
QoE	Quality of Experience
QoS	Quality of Services
R	
RACH	Random Access Channels
RAT	Radio Access Technology
RL	Reinforcement Learning
RMSE	Root Mean Square Error
RN	Relay Node
RRU	Radio Resource Utilization
RSS	Rich Site Summary (also known as Real Simple Syndication)
RSSI	Received Signal Strength Indicator
S	
SOM	Self-Organizing Map
SP	Service Provider
T	
U	

UE	User Equipment
V	
W	
X	
Y	
Z	

11 APPENDIX B – LIST OF PUBLICATIONS (JUNE 2015)

Short CV	
Aimilia Bantouna was born in Athens, Greece, in 1984. She has received the bachelor's degree from the Department of Applied Mathematics and Physics of the National Technical University of Athens, Greece, in 2007 and the M.Sc. diploma from the Department of Digital Systems of the University of Piraeus, Greece, in 2010. Since June 2011 she is research engineer at the University of Piraeus, in the Laboratory of Telecommunication Networks and Integrated Services. She has been involved in EU-funded FP7/ICT Univerself Integrated Project (2010-2013) and in the EU-funded FP7/ICT ACROPOLIS (Advanced Coexistence technologies for Radio Optimization in Licensed and unlicensed Spectrum) Network of Excellence (2010-2013) which were funded by the European Commission under 7th Framework Programmes (FP7). Her main focus is on building knowledge with respect to network and user behavior through machine learning techniques (especially unsupervised ones) in order to be used for predicting upcoming situations of the network.	
Journal Publications	
1.	K. Tsagkaris, G. Nguengang, A. Galani, I. Gridabenyahia, M. Ghader, A. Kaloxylos, M. Gruber, A. Kousaridas, M. Bouet, S. Georgoulas, A. Bantouna, N. Alonistioti, P. Demestichas, "A survey of autonomic networking architectures: towards a Unified Management Framework", International Journal of Network Management, Oct. 2013, DOI: 10.1002/nem.1841

2.	A. Bantouna, G. Poulios, K. Tsagkaris, P. Demestichas, "Network Load Predictions based on Big Data and the Utilization of Self-Organizing Maps", JNSM Special Issue 2013 : Springer Journal of Network and Systems Management – Special Issue on "Data Mining for Monitoring and Managing Systems and Networks", Sept. 2013, DOI: 10.1007/s10922-013-9285-1, Volume 22, Issue 2 (2014), Page 150-173
3.	L. Ciavaglia, S. Ghamri - Doudane, M. Smirnov, P. Demestichas, V. Stavroulaki, A. Bantouna, Unifying Management of Future Networks with Trust, Bell Labs Technical Journal (BLTJ), Special Issue On "Delivering Network Assurance through Secure and Reliable Products, Software, Services and Solutions", Vol. 17, No. 3, p. 193-212, December 2012
4.	A. Bantouna, V. Stavroulaki, Y. Kritikou, K. Tsagkaris, P. Demestichas, K. Moessner, "An overview of learning mechanisms for cognitive systems", Published to EURASIP Special Issue on Ten Years of Cognitive Radio: State of the Art and Perspectives, EURASIP Journal on Wireless Communications and Networking 2012, 2012:22 doi:10.1186/1687-1499-2012-22, January 2012.
5.	V. Stavroulaki, A. Bantouna, Y. Kritikou, K. Tsagkaris, P. Demestichas, P. Blasco, F. Bader, M. Dohler, D. Denkovski, V. Atanasovski, L. Gavrilovska, K. Moessner, "Knowledge Management Toolbox: Machine Learning for Cognitive Radio Network", Published in IEEE Vehicular Technology Magazine, Special Issue on "Applications of Cognitive Radio Networks", vol.7, no.2, pp.91-99, June 2012 doi: 10.1109/MVT.2012.2190196
6.	K. Tsagkaris, A. Bantouna, P. Demestichas, "Self-Organizing Maps for Advanced Learning in Cognitive Radio Systems", Computers & Electrical Engineering, Elsevier, Vol. 38, No. 4, p. 862–881, July 2012, http://dx.doi.org/10.1016/j.compeleceng.2012.03.008

Book Chapters	
1.	A. Bantouna, K. Tsagkaris, V. Stavroulaki, P. Demestichas, G. Poulios, "Machine Learning applied to Cognitive Communications", Cognitive Communications: Distributed Artificial Intelligence (DAI), Regulatory Policy & Economics, Implementation. H. Zhang and D. Grace, J. Wiley and Sons, Ltd, Chichester, UK, Print ISBN: 9781119951506 (Oct. 2012), Online ISBN: 9781118360316 (Jul. 2012), chapter 6, doi: 10.1002/9781118360316.ch6
2.	A. Bantouna, K. Tsagkaris, V. Stavroulaki, G. Poulios, P. Demestichas, "Learning Techniques for Context Diagnosis and Prediction in Cognitive Communications", Cognitive Communications: Distributed Artificial Intelligence (DAI), Regulatory Policy & Economics, Implementation. H. Zhang and D. Grace, J. Wiley and Sons, Ltd, Chichester, UK, Print ISBN: 9781119951506 (Oct. 2012), Online ISBN: 9781118360316 (Jul. 2012), chapter 9, doi: 10 10.1002/9781118360316.ch9
Conference Publications	
1.	A. Bantouna, K. Tsagkaris, P. Demestichas, "Knowledge Functional Block for E-UTRAN", Accepted to 5th IEEE International Workshop on Management of Emerging Networks and Services (IEEE MENS 2013), Atlanta , GA USA, Dec. 2013
2.	Oliver Holland, Hamid Aghvami, Sylwia Romaszko, Daniel Denkovski, Valentina Pavlovska, Liljana Gavrilovska, Aimilia Bantouna, Vera Stavroulaki, Yiouli Kritikou, Panagiotis Demestichas, Eduard Jorswieck, "The ICT-ACROPOLIS Network of Excellence: Spectrum Sharing and Coexistence based on the Cognition Cycle", Future Network & Mobile Summit 2013 (FUNEMS 2013), Lisbon, Portugal, July 3-4, 2013

3.	L. De Nardis, M.-G. Di Benedetto, O. Holland, A. Akhtar, H. Aghvami, V. Rakovic, V. Atanasovski, L. Gavrilovska, V. Stavroulaki, Y. Kritikou, A. Bantouna, P. Demestichas, D. Tassetto, S. Bovelli, S. Romaszko, "Neighbour and network discovery in cognitive radio networks: research activities and results in the ACROPOLIS Network of Excellence", In Proc. of the 19th European Wireless Conference (EW 2013), Guildford, UK, 16-18 April 2013
4.	L. de Nardis, M. G. di Benedetto, V. Stavroulaki, A. Bantouna, Y. Kritikou, P. Demestichas, "Role of neighbour discovery in distributed learning and knowledge sharing algorithms for cognitive wireless networks", In Proc. of the International Symposium on Wireless Communication Systems (ISWCS) 2012, Special Session on Advances on Cognitive Radio and Learning Mechanisms, Paris, France, 28-31 August 2012, p. 421-425, DOI: 10.1109/ISWCS.2012.6328402
5.	A. Georgakopoulos, P. Demestichas, V. Stavroulaki, A. Bantouna, "Mechanisms for Information and Knowledge Sharing in Wireless Communication Systems", Accepted to the International Symposium on Wireless Communication Systems (ISWCS) 2012, Special Session on Advances on Cognitive Radio and Learning Mechanisms, Paris, France, 28-31 August 2012.
6.	A. Bantouna, K. Tsagkaris, G. Poulios, A. Manzalini, P. Demestichas, "Knowledge in Support of Congestion Control Mechanisms", poster, Future Network Mobile Summit (FuNeMS) 2012, 4 - 6 July 2012, Berlin, Germany
7.	M. Ghader, A. Bantouna, L. Bennacer, G. Calochira, B. Fuentes, G. Katsikas, Z. Yousaf, "On Accomplishing Context Awareness for autonomic network management", accepted at Future Network and Mobile Summit 2012, 4 - 6 July 2012, Berlin, Germany

8.	P. Spapis, R. Razavi, S. Georgoulas, Z. Altman, R. Combes, A. Bantouna, "On the role of learning in autonomic network management: the UniverSelf project approach", accepted at Future Network and Mobile Summit 2012, 4 - 6 July 2012, Berlin, Germany
9.	K. Tsagkaris, P. Vlacheas, A. Bantouna, P. Demestichas, G. Nguengang, M. Bouet, L. Ciavaglia, P. Peloso, I. Grida BenYahia, C. Destré "Operator-driven Framework for Establishing and Unifying Autonomic Network and Service Management Solutions", in proceedings of the 3rd IEEE International Workshop on Management of Emerging Networks and Services (MENS 2011) at the IEEE Global Communications Conference (GLOBECOM 2011), December 2011, Houston, USA
10.	A. Bantouna (presenter), K. Tsagkaris, V. Stavroulaki, P. Demestichas, "Machine Learning Techniques for Autonomic/ Cognitive Networking", in the context of the 7th International Conference on Network and Service Management CNSM 2011, held on 24-28 October 2011, in Paris, France
11.	A. Bantouna, K. Tsagkaris, P. Demestichas, "Self-Organizing Maps for Improved Learning in Cognitive Radio Systems", In Proc. 1st International Conference for Undergraduate and Postgraduate Students in Computer Engineering, Informatics, related Technologies and Applications 2010 (Eureka! 2010), Patras, Greece, October 2010, International Conference Papers
12.	A. Bantouna, K. Tsagkaris, P. Demestichas, "Self-Organizing Maps for improving the channel estimation and predictive modelling phase of cognitive radio systems", In Proc. 20th International Conference on Artificial Neural Networks (ICANN 2010), Thessaloniki, Greece, September 2010, International Conference Papers

Workshop Publications	
1.	A. Bantouna, G. Poulios, K. Tsagkaris, P. Demestichas, "Dynamic Management of Cognitive Radio Networks using Big Data", 3rd ACROPOLIS Workshop and Industry Panel, London, Sept. 2013
2.	Luca De Nardis, Nikos Dimitriou, Andreas Zalonis, Aimilia Bantouna, Vera Stavroulaki, Kostas Tsagkaris, Panagiotis Demestichas, Adrian Kliks, Dionysia Triantafyllopoulou, Ricardo Blasco-Serrano, White Paper: The Network and Upper Layers within Cognitive Radio - Context Identification and Decision Making Aspects
3.	L. De Nardis, M.-G. Di Benedetto, V. Stavroulaki, A. Bantouna, Y. Kritikou, P. Demestichas, "Role of neighbor discovery in distributed learning and knowledge sharing algorithms for cognitive wireless networks", ICT-ACROPOLIS First Annual Workshop, Castelldefels - Barcelona, Spain, October 2011.
4.	A. Bantouna, Y. Kritikou, V. Stavroulaki, P. Demestichas, K. Moessner, "Learning in Cognitive Systems: An Overview of Basic Mechanisms and Implementation Approaches", ICT-ACROPOLIS First Annual Workshop, Castelldefels - Barcelona, Spain, October 2011.