Performance Anomaly Detection in 802.11 Wireless Networks Applying Hidden Markov Models

Author:
Anisa ALLAHDADI

Supervisor:
Prof. Ricardo MORLA
Co-Supervisor:
Prof. Jaime S. CARDOSO

Head of Jury:
Prof. Gabriel DAVID

Jury Member:
Prof. João P. VILELA
Jury Member:
Prof. João Paulo CARVALHO
Jury Member:
Prof. Paulo SALVADOR
Jury Member:
Prof. Ricardo MORLA

Doctoral Program in Computer Science of the Universities of Minho, Aveiro and Porto

February 12, 2020

DOCTORAL THESIS

Performance Anomaly Detection in 802.11 Wireless Networks Applying Hidden Markov Models

Author:
Anisa ALLAHDADI

Supervisor:
Prof. Ricardo MORLA

Co-Supervisor:
Prof. Jaime S. CARDOSO

Thesis submitted to Faculty of Sciences of the University of Porto for the Doctor Degree in Computer Science within the Joint Doctoral Program in Computer Science of the Universities of Minho, Aveiro and Porto

February 12, 2020

"Knowledge is as wings to man's life, and a ladder for his ascent. Its acquisition is incumbent upon everyone. The knowledge of such sciences, however, should be acquired as can profit the peoples of the earth, and not those which begin with words and end with words. Great indeed is the claim of scientists and craftsmen on the peoples of the world.... In truth, knowledge is a veritable treasure for man, and a source of glory, of bounty, of joy, of exaltation, of cheer and gladness unto him."

Bahá'u'lláh

Abstract

IEEE 802.11 Wireless Networks are getting increasingly popular providing ubiquitous Internet access for a large number of users in university campuses, enterprises, urban areas, and many other public places. Among the many characteristics of these large-scale 802.11 networks is the transition of huge volumes of traffic as a result of intensive usage from different locations in the wireless covered area. The mobile users often suffer from connectivity problems and performance issues due to unstable radio conditions and dynamic user behavior among other reasons. Anomaly detection and distinction are among the major challenges that network managers encounter. Monitoring the broad and complex Wireless Local Area Networks (WLANs) often requires heavy instrumentation of the user devices, which makes the anomaly detection analysis even harder.

In this thesis we propose to use Hidden Markov Models (HMMs) and its variation models: 1) to inspect and characterize the dynamic usage pattern of wireless networks in terms of the changing traffic patterns, mobility of the users and the anomalies, 2) to model usage behaviors of individual access points (APs) or groups of APs, 3) to introduce and improve anomaly detection techniques based on the temporal data sequences, and 4) to represent the spatio-temporal anomaly detection approaches based on the additional spatial information.

To achieve this purpose, we start by performing an exhaustive outlier detection analysis using the state of the art methodologies as well as the proposed HMM modeling approach. We explore and compare individual HMMs versus single HMM and mixture of HMMs. We further present a number of network anomalous patterns based on HMM parameters like hidden states' transition and partial likelihood of the observation sequences.

In order to understand how to improve the anomaly detection results, we then propose Gaussian Mixture Models (GMMs) and Hidden Markov Models (HMMs) as time-invariant and time-variant modeling techniques, respectively. We represent the anomaly detection techniques by GMM and HMM and analyze the root causes of the anomalies. We further improve the HMM models using Universal Background Model (UBM) as a robust technique to initialize HMMs using all the available data. We represent the application of anomaly detection in three different aspects: 1) detection of anomalous time-series in a database of time-series, 2) distinction of anomalous patterns, and 3) detection of anomalous points within a given time-series.

Finally, we try to understand how and to which extent the anomaly detection results can improve by adding spatial information. In particular, we apply a hybrid integration of the Self-Organizing Map (SOM) and the Hidden Markov Model (HMM), called SOHMMM for spatio-temporal anomaly detection in 802.11 wireless network. We believe that while modeling an independent HMM per AP misses the opportunity to explore similarities between APs for improvement of learning, a single HMM for all APs loses the flexibility to learn AP specific behaviour. The UBM-HMM is an improvement in the right direction but the relations between APs are only used in the initial phase, where one learns the UBM to then initialize the individual HMM models per AP. In SOHMMM we focus on actual proximity of APs as a determinant factor in connectivity and performance problems and employ SOHMMM to exploit the semantic connectivity between adjacent HMMs. We further extend the online gradient descent unsupervised learning algorithm of SOHMMM for multivariate Gaussian emissions.

We validate our proposed methodologies on three main data sets for experimental analysis and evaluation purposes: 1) RADIUS authentication log data collected

at the hotspot of the Faculty of Engineering of the University of Porto (FEUP) that summarizes the connection of more than 45 thousands users to 364 APs in over two years, 2) an exploratory small testbed deployed in FreeRADIUS server and at a home environment with 6 wireless users connected to 1 AP from heterogeneous devices, 3) a middle-sized wireless network simulation containing 100 wireless users associated to 10 APs located in a wireless ground, simulated in a $1500m \times 1200m$ OMNeT++ wireless simulator and INET framework.

Our findings indicate that: 1) the single HMM and mixture HMM models outperforms the individual HMMs in terms of accuracy and HMM indicators conformity (Chapter 4), 2) HMM as time-variant approach outperforms GMM as time-invariant approach in obtaining higher detection ratio while producing minor false alarms (Chapter 5), 3) HMM and HMM-UBM models are both capable of detecting greater proportion of anomalies while producing only a small false positive ratio, compared to baseline approaches like RawData and PCA (Chapter 5), 4) SOHMMM algorithm can improve the anomaly detection accuracy and sensitivity compared to HMM-UBM and Z-SOHMMM (SOHMMM with zero neighborhood) techniques in various anomalous scenarios (Chapter 6).

Resumo

As redes sem fio IEEE 802.11 estão a tornar-se cada vez mais populares fornecendo acesso ubíquo à Internet para um grande número de usuários em campus universitários, empresas, áreas urbanas e muitos outros locais públicos. Entre as muitas características dessas redes 802.11 de larga escala está a transição de grandes volumes de tráfego como resultado do uso intensivo de diferentes locais na área coberta sem fio. Os utilizadores móveis geralmente sofrem com problemas de conectividade e problemas de desempenho devido a condições de rádio instáveis e comportamentos dinâmicos do utilizador, entre outras razões. A detecção e distinção de anomalias estão entre os principais desafios que os gerentes de rede enfrentam. A monitorização de amplas e complexas redes locais sem fio (WLANs) geralmente requer instrumentação pesada dos dispositivos do utilizador, o que dificulta ainda mais a análise da detecção de anomalias.

Nesta tese propomos o uso de Modelos Ocultos de Markov (HMMs) e seus modelos de variação, para: 1) inspecionar e caracterizar o padrão de uso dinâmico de redes sem fio em termos de mudança de padrões de tráfego, mobilidade dos utilizadores e das anomalias, 2) modelar comportamentos de uso de pontos de acesso individuais (APs) ou grupos de APs, 3) introduzir e melhorar técnicas de detecção de anomalias baseadas nas sequências de dados temporais e, 4) representar as abordagens de detecção de anomalias espacio-temporais baseadas na informação espacial adicional.

Para atingir este objetivo, começamos por realizar uma análise exaustiva da detecção de outliers usando as metodologias do estado da arte, assim como a abordagem de modelação de HMM proposta. Exploramos e comparamos HMMs individuais versus um HMM simples e a mistura de HMMs. Além disso, apresentamos vários padrões anômalos de rede baseados em parâmetros HMM, como a transição de estados ocultos e a verossimilhança parcial das seqüências de observação.

Para entender como melhorar os resultados de detecção de anomalias, propomos Modelos de Mistura Gaussiana (GMMs) e Modelos Ocultos de Markov (HMMs) como técnicas de modelação invariantes no tempo e variantes no tempo, respectivamente. Representamos as técnicas de detecção de anomalias por GMM e HMM e analisamos as causas raízes das anomalias. Melhoramos ainda mais os modelos HMM usando o Universal Background Model (UBM) como uma técnica robusta para inicializar HMMs usando todos os dados disponíveis. Representamos a aplicação da detecção de anomalias em três aspectos diferentes: 1) detecção de séries temporais anômalas em um banco de dados de séries temporais, 2) distinção de padrões anômalos e 3) detecção de pontos anômalos dentro de uma determinada série temporal.

Finalmente, tentamos entender como e em que medida os resultados da detecção de anomalias podem melhorar, adicionando informações espaciais. Em particular, aplicamos uma integração híbrida do Mapa Auto-Organizável (SOM) e do Modelo Oculto de Markov (HMM), chamado SOHMMM para detecção de anomalia espaço-temporal na rede sem fio 802.11. Acreditamos que, embora a modelação de um HMM independente por AP perca a oportunidade de explorar semelhanças entre os APs para melhorar a aprendizagem, um único HMM para todos os APs perde a flexibilidade de aprender o comportamento específico do AP. O UBM-HMM é uma melhoria na direção certa, mas as relações entre os APs são usadas apenas na fase inicial, onde se aprende o UBM para, então, inicializar os modelos HMM individuais por AP. No SOHMMM, nos concentramos na proximidade real dos APs como

um fator determinante dos problemas de conectividade e desempenho e empregamos o SOHMMM para explorar a conectividade semântica entre os HMMs adjacentes. Além disso, estendemos o algoritmo de aprendizado não supervisionado de SOHMMM de gradiente de descida on-line para emissões gaussianas multivariadas. Validamos as nossas metodologias propostas em três conjuntos de dados principais para análise experimental e avaliação: 1) dados do registo de autenticação RADIUS recolhidos no hotspot da Faculdade de Engenharia da Universidade do Porto (FEUP) que resume a ligação de mais de 45 mil utilizadores para 364 APs em mais de dois anos, 2) um pequeno testbed exploratório implantado no servidor FreeRADIUS e em um ambiente doméstico com 6 usuários wireless conectados a 1 AP de dispositivos heterogêneos, 3) uma simulação de rede wireless de tamanho médio contendo 100 usuários wireless associados 10 APs localizados em um terreno sem fio, simulados em um simulador sem fio OMNeT ++ de $1500m \times 1200m$ e uma estrutura INET.

Nossas descobertas indicam que: 1) os modelos HMM e HMM de mistura única superam os HMMs individuais em termos de precisão e conformidade dos indicadores HMM (Capítulo 4); 2) HMM como abordagem de variante de tempo supera o GMM como abordagem invariante no tempo para obter maior taxa de detecção enquanto produz menos falsos alarmes (Capítulo 5), 3) os modelos HMM e HMM-UBM são capazes de detectar uma maior proporção de anomalias enquanto produzem apenas uma pequena proporção de falsos positivos, em comparação com abordagens de base como RawData e PCA (Capítulo 5), 4) O algoritmo SOHMMM pode melhorar a precisão e a sensibilidade da detecção de anomalias em comparação com as técnicas HMM-UBM e Z-SOHMMM (SOHMMM com vizinhança zero) em vários cenários anômalos (Capítulo 6).

Acknowledgements

First, I would like to express my sincere gratitude to my Ph.D. supervisor Professor Ricardo Morla, for his continuous support, motivating discussion, profound understanding, and precious insights. His guidance helped me in all the stages of research and writing of this thesis. I would also like to convey my sincere gratitude to my Ph.D. co-supervisor Professor Jaime S. Cardoso who despite his busy schedule accepted to be my co-supervisor and always provided me with innovative ideas, insightful advice, and valuable comments.

I would like to thank the Foundation for Science and Technology (FCT) in Portugal for their financial support throughout my Ph.D. studies. Assuredly, without their support the results of this work would not be the same. FCT support was received through grant number SFRH/BD/99714/2014. Further, I am very grateful to the directors of INESC-TEC for allowing me to conduct my research in their highly reputable research laboratory (CTM). I received a data set from FEUP which I used in the initial phases of my Ph.D. research. Moreover, I am very thankful to my instructors and colleagues in the MAP-i Ph.D. Programme.

I would like to express my heartfelt gratitude to BIHE for giving me the opportunity to pursue my studies and research, at a time when Bahá'í scholars are otherwised barred from receiving an academic education in Iran, due to their beliefs. A special thanks to Mr. Kamran Mortezaie and Mr. Mahmoud Badavam, who spent years in prison for the sake of educating Bahá'í students, and Mrs. Azita Rafizadeh who was my former instructor and in the time of writing this thesis is still in prison for the same reason. I would like to convey my sincere gratitude to all my instructors and colleagues at BIHE for having been a source of inspiration my entire life, as well as for all their supports and devotions.

I am extremely grateful to my parents and my brother for their support, encouragement and all their sacrifices throughout my life that made this long journey possible for me. Also, I am grateful to my in-laws for their cooperation and support.

Last but not least, I would like to thank my beloved husband, Peyman Sazedj, for all his patience, his constant support and sacrifices. He was always present in my most challenging moments, pushing me forward, cheering me up, and paving the way for my progress and happiness. I am immensely thankful to our little son, Taraz, who has always been a source of joy and laughter to us, and shines like a brilliant star in the sky of our lives. And also to my new baby girl, Aliya, whose arrival in this world coincided with the accomplishment of this journey.

UNIÃO EUROPEIA
Fundo Social Europeu

FCT Fundação para a Ciência e a Tecnologia

INESCTEC TECHNOLOGY & SCIENCE

MAPi DOCTORAL PROGRAMME IN COMPUTER SCIENCE

Contents

List of Figures

List of Tables

List of Abbreviations

AAA	**A**uthentication **A**uthorization **A**ccounting
AP	**A**ccess **P**oint
AUC	**A**rea **U**nder **C**urve
BSS	**B**asic **S**ervice **S**et
CDF	**C**umulative **D**istribution **F**unction
DES	**D**iscrete **E**vent **S**imulator
EM	**E**xpectation **M**aximization
ESS	**E**xtended **S**ervice **S**et
ESSID	**E**xtended **S**ervice **S**et **ID**entification
FPR	**F**alse **P**ositive **R**ate
FEUP	**F**aculty of **E**ngineering of the **U**niversity of **P**orto
GMM	**G**aussian **M**ixture **M**odel
HMM	**H**idden **M**arkov **M**odel
IBSS	**I**ndependent **B**asic **S**ervice **S**et
IEEE	**I**nstitute of **E**lectrical and **E**lectronics **E**ngineers
IETF	**I**nternet **E**ngineering **T**ask **F**orce
ISP	**I**nternet **S**ervice **P**rovider
KL	**K**ullback **L**eibler
LAN	**L**ocal **A**rea **N**etwork
LL	**L**og-likelihood
MAP	**M**aximum **A** **P**osteriori
MC	**M**onte **C**arlo

PDF	**P**robability **D**ensity **F**unction
PCA	**P**rincipal **C**omponent **A**nalysis
PHY	**P**hysical **L**ayer
QoS	**Q**uality **o**f **S**ervice
RADIUS	**R**emote **A**uthentication **D**ial **I**n **S**ervice
RF	**R**adio **F**requency
RFC	**R**equest **F**or **C**omments
ROC	**R**eceiver **O**perating **C**haracteristic
RSS	**R**eceived **S**ignal **S**trength
RSSI	**R**eceived **S**ignal **S**trength **I**ndicator
SNMP	**S**imple **N**etwork **M**anagement **P**rotocol
SOM	**S**elf **O**rganizing **M**ap
SOHMMM	**S**elf **O**rganizing **H**idden **M**arkov **M**odel **M**ap
STA	Wireless **Sta**tion
TCP	**T**ransmission **C**ontrol **P**rotocol
TPR	**T**rue **P**ositive **R**ate
UBM	**U**niversal **B**ackground **M**odel
UDP	**U**ser **D**atagram **P**rotocol
WLAN	**W**ireless **L**ocal **A**rea **N**etwork
WPA	**W**i-Fi **P**rotected **A**ccess
WEP	**W**ired **E**quivalent **P**rivacy

Chapter 1

Introduction

Recently, the mass deployment of communication devices and wireless infrastructures has been observed extensively. As the employment of these technologies becomes an important part of peoples' lives, utilizing these devices has the potential ability of reflecting the lifestyle of human beings or the specific features of their inhabitant places. Understanding the user's behavioral patterns will also play an important role in dealing with network management issues to provide higher quality of service, preventing network failures due to congestion, predicting traffic flow, designing behavior-aware applications among others.

Wireless LANs (WLANs) with 802.11 technologies are getting deployed principally at most university campuses, enterprises and organizations, shopping centers, airports and in so many other public places. These large-scale networks, particularly speaking of IEEE 802.11 Infrastructure mode, consist of basic network components: Wireless Stations, wired stations, and the Access Points (AP) that function as connection links between the wired and wireless sections. An AP in a wireless network establishes wireless connectivity with a wireless station within WLAN and supports a particular number of devices at a given time. The APs provide coverage and capacity for supporting mobile clients with heterogeneous devices and a variety of applications. The number of connected devices sustained by an AP varies over time since new devices come in and existing devices leave by. The type of services utilized and the amount of traffic transferred also affect the number of supported devices and hence the quality of service provided. Figure 1.1 shows the schema of a typical WLAN and its main components.

Among the many characteristics of these large-scale networks is the transmission of huge volumes of traffic as a result of intensive usage from different locations in the wireless area. The mobile clients demand reliable connections and high performance in all circumstances and expect their applications to work smoothly around the wireless covered field, but this is an ideal case which is not always achievable. Most of the time, wireless users suffer from low coverage, intermittent connectivity, authentication failure, degraded performance and many other complications originated from the unreliable nature of the wireless connection and dynamic usage pattern of other users in the vicinity. The question of performance becomes increasingly important as new applications demand sufficient bandwidth and reliable medium access. Detection of aforementioned problems that affect the performance of the individual users in the network is of great importance for network managers.

One of the objectives of this work is to inspect and characterize the usage pattern of the wireless networks and its inherent dynamics by exploring the spatial proximity of access points as well as their timely usage pattern, and to provide robust models for anomaly detection.

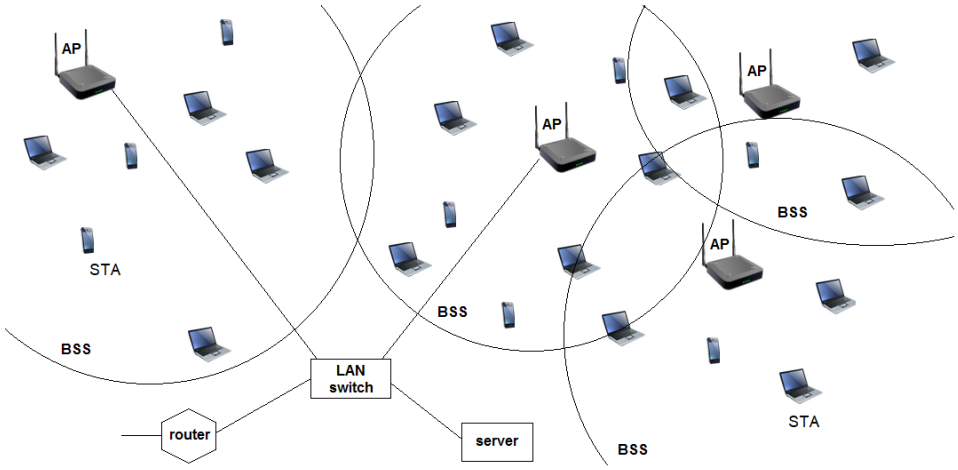


FIGURE 1.1: WLAN APs and the coverage area of the wireless stations (STA)

In the following section, we elaborate some details of wireless setup in infrastructure mode, and explain how a wireless station associate to an access point. We further provide a brief description of RADIUS protocol and the process of authentication and authorization of users to the network under this protocol. This content is important so the reader can better understand not only the wireless performance issues, but also the data in which our solution is based.

The remaining sections of this chapter present performance issues for wireless users in subsection 1.2, challenges of the wireless network management in subsection 1.3, proposed solution that contain usage modeling and anomaly detection in subsection 1.4, the most important research questions of the current work in subsection 1.5, the key contributions in subsection 1.6, and thesis structure in subsection 1.7.

1.1 Wireless Setup in Infrastructure Mode

1.1.1 Association of Wireless Station to Access Point

The process of the association of a wireless mobile station to an AP, as it is currently implemented by most manufacturers is described as follows: A wireless station scans the available channels of each AP in the neighborhood and listens to the beacon (passive approach) or probe response frames (active approach). IEEE 802.11 protocol defines a number of Wi-Fi channels ranging from 2.4 GHz to 5.9 GHz. The Wi-Fi channels that are the concern of this work (802.11 b/g/n) are listed in the 2.4 GHz range and consist of one to eleven (up to fourteen in some countries) channels. The wireless station stores the received signal strength indicator (RSSI) of the APs in the vicinity and other relevant information such as extended service set identification (ESSID), encryption type (e.g. WPA, WEP), etc. When the scanning process is over, the wireless station typically selects an AP with the highest RSSI among the observed APs in its proximity. After the process of authentication/authorization is accomplished, the permission is granted to the wireless station and the connection is established. Forthwith, the wireless station is associated with the new AP and the user is ready to send and receive traffic through that AP. The wireless station will be dis-associated from the current AP under the mobility circumstances, AP shutdown

or halt, RSSI recession or some other normal or abnormal consequences of network fluctuations. The process of AP selection only based on the strongest RSSID lead to load imbalance problem, while some APs are overcrowded and the other available APs remain idle.

1.1.2 Remote Authentication Dial-In User Service (RADIUS)

Remote Authentication Dial-In User Service (RADIUS) is a network protocol that enables remote access servers to communicate with a central server to authenticate dial-in users and authorize their access to the requested system or service. RADIUS is commonly used by Internet Service Providers (ISPs), cellular network providers, and corporate and educational networks, and it allows the management of user profiles in a central database that all remote servers can share. Having a central service facilitates the process of tracking usage for billing and network statistics. RADIUS is a de facto industry standard used by a number of network product companies and it is a proposed IETF standard [1]. This protocol is used to provide network authentication, authorization, and accounting services, and it is particularly described in Request for Comments (RFC) 2865 [2] and RFC 2866 [3].

According to RADIUS protocol, whenever a client associates to an 802.11 AP, a log event "START" is recorded in the accounting database. While the client is still connected to this AP, every 10 or 15 minutes (based on the server configuration) an interim log event "ALIVE" is issued to refresh the connection between the client and the AP. Eventually, when the user decides to disconnect from the network, or for some reason it is forced to leave the network, a log event "STOP" is recorded, which marks the end of the association period of this user.

RADIUS serves three main purposes as follows:

- Authenticates users before granting them access to the network.
- Authorizes the authenticated users for specific network services.
- Accounts the usage activity of the authorized users for the services in use.

AAA stands for "Authentication, Authorization, and Accounting". It defines an architecture that authenticates and grants authorization to users and accounts for their activity. When AAA is not used, the architecture is described as "open", where anyone can gain access and do anything, without any tracking.

A Network Access Server (NAS) operates as a client of the RADIUS accounting server. The client is responsible for passing user accounting information to a designated RADIUS accounting server. The RADIUS accounting server is in charge of receiving the accounting request and returning a response to the client indicating that it has successfully received the request. The RADIUS accounting server can act as a proxy client to other kinds of accounting servers [3].

Accounting

Accounting refers to the recording of resources users consume during the time they are connected to the network. The information gathered can include the total system time used, and the amount of data sent or received by the user during a session. Over a network session, the NAS periodically sends an accounting data of user activity to the server (in "Alive" or "Stop" sessions). This data is mainly used for the billing purposes. However, we used the accounting information for the reason of

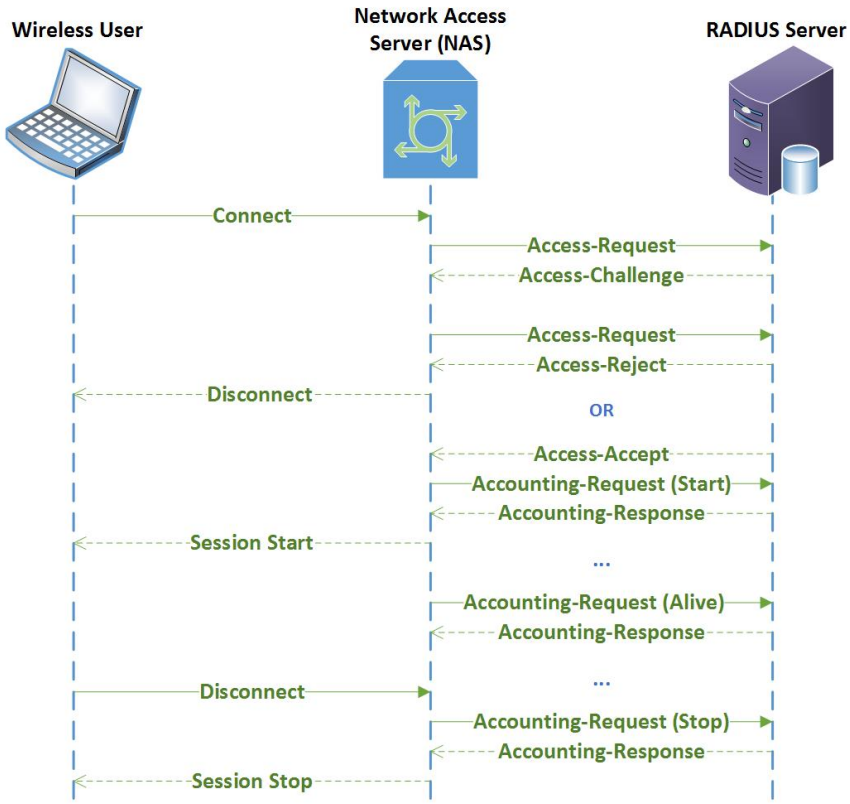


FIGURE 1.2: The Authentication and Authorization Process in RADIUS

network monitoring and management as the log data set is already stored in a central database, the RADIUS server, and facilitates the data collection process.

The detailed information of users' activities is not included in the summary sent by NAS- for instance the visited web sites or particular protocols in use is local to the NAS- and is not available to the RADIUS server. Transactions between the client and RADIUS server are authenticated through the use of a shared secret, which is never sent over the network. In addition, user passwords are sent encrypted between the client and RADIUS server to eliminate the possibility of snooping on an insecure network.

The process of authentication and authorization is delineated in Figure 1.2.

1.2 Performance Issues for Wireless Users

Having further explored the connectivity procedure in wireless networks, some inherent concerns and dilemmas become more clear. In Wireless 802.11 networks, mobile stations perform an active or passive scanning process to discover available APs in the vicinity and connect to an AP with the highest received signal strength (RSS) [4]. This association strategy, only based on RSS, can lead to many connectivity problems and performance issues as it may result in significant load imbalance between APs. The overloaded APs can still present high RSS and try to accommodate more stations while other APs are only slightly loaded or even idle.

Another source of performance degradation in WLANs is the multi-rate flexibility and the fairness mechanism of the MAC protocol, when a station far from the

AP reduces its bit rate to avoid repeated unsuccessful frame transmission and as a result, degrades the throughput of the other stations associated with the same AP [5].

In addition to the aforementioned problems, due to the unreliable and time-varying nature of the wireless channels, 802.11 networks usually suffer from many deficiencies such as exposed and hidden terminals, capture effect, interferences, signal fading, inconsistent coverage among others. In such circumstances, high packet loss is observed [6] that results in inconsistent connectivity and low performance. Additionally, the traffic load and users movement on different access points in the wireless covered area vary from time to time, making these network management tasks even harder. Network managers are concerned about discovering such sort of problems and abnormal events occurring in their network. Detection of anomalies is not only advantageous for prompting immediate administrative actions but also useful for long-term network design, planning, and maintenance decisions as the network infrastructure and usage evolve over time. This paves the way for gathering data from existing mechanisms in the network; in our case we focus on the RADIUS accounting data.

1.3 Challenges of the Wireless Network Management

In large deployments of 802.11 networks with varying usage, channel conditions, and operational constraints, network managers often demand tools that provide them with a comprehensive view of the entire network for automatic detection of the problems. In such widespread networks, where at any moment there is a high possibility of malfunctioning of APs and user devices, the necessity of such automatic tools or applications is vital to preserve the quality of service at an acceptable level.

Monitoring the infrastructure by any means rather than intelligent diagnostic tools seems inconvenient in practice or overpriced in budget. For example, it is expensive to deploy third party devices like sensors and sniffers individually on clients machines or APs for detection of problems in different OSI layers, as studied earlier [7, 8, 9]. And it seems impractical for network staff to walk around the wireless covered area with a device in their hand monitoring the network and measuring the quality of connections at any time.

1.4 Proposed Solution: Usage Modeling and Anomaly Detection

We propose to use Hidden Markov Models (HMMs) and its variation models: 1) to inspect and characterize the dynamic usage pattern of wireless networks in terms of the changing traffic patterns, mobility of the users and the anomalies, 2) to model usage behaviors of individual access points (APs) or groups of APs, 3) to introduce and improve anomaly detection techniques based on the temporal data sequences, and 4) to represent the spatio-temporal anomaly detection approaches based on the additional spatial information. The employed methodology is based on the development of HMM models and a detection tool using Wi-Fi campus data, Testbed deployment and wireless simulation.

An exhaustive analysis is performed for outlier detection in 802.11 wireless networks using the state of the art methodologies in addition to the proposed HMM

techniques. Chapter 4 has taken this approach into account. Furthermore a number of network anomalous patterns are represented considering HMM parameters such as hidden states' transition and partial likelihood of the observation sequences. Moreover, individual HMMs versus single HMM (one for all APs) and mixture of HMMs (groups of HMM) are investigated.

In Chapter 5, the AP usage data of 802.11 WLAN is analyzed and anomaly detection techniques for AP level anomalous events are proposed. Gaussian Mixture Models (GMMs) as time-invariant and Hidden Markov Models (HMMs) as time-variant modeling techniques are presented, and a case study on FEUP data set (3.2) is performed to inspect these two methodologies. Further, the anomaly detection techniques by GMM are represented as distant data points that hardly belong to any Gaussian component, and by HMM as data points with the minimum likelihood values. Some of the root cause of the low likelihood values are analyzed as divergence from the assigned hidden states as well as the low probability in state transition. The experiments are carried out on an exploratory Testbed deployed in a home environment (3.3). In the second part of this chapter, Universal Background Modeling (UBM) approach is introduced for a robust initialization of the HMMs using the data available from all experiments regardless of containing anomalies or not. Regarding the anomaly detection techniques three main approaches are considered: 1) detection of anomalous time-series in a database of time-series, 2) distinction of anomalous patterns, and 3) detection of anomalous points within a given time-series. For the evaluation part a number of anomalous scenarios are simulated in OMNeT++/INET for 5 APs and 30 STAs (smaller version of wireless network simulation in 3.4). Then the detection results of HMM and HMM-UBM techniques are evaluated versus the baseline approaches, namely RawData and PCA.

In Chapter 6 a hybrid integration of the Self-Organizing Map (SOM) and the Hidden Markov Model (HMM) is applied, called SOHMMM for anomaly detection in 802.11 wireless network. Further the online gradient descent unsupervised learning algorithm of SOHMMM is extended for multivariate Gaussian emissions. Experimental analysis consists of two main parts: synthetic data, and wireless simulation data (3.4). In synthetic data analysis the distance between randomly initialized HMMs and reference HMMs are estimated to investigate whether the random HMMs converge to the reference HMMs efficiently. The experiments are further repeated for various observation sequence lengths and different neighborhood sizes. Wireless Simulation data analysis display how the SOHMMM algorithm improve the anomaly detection accuracy and sensitivity compared to HMM-UBM and Z-SOHMMM techniques in AP shutdown/halt, AP overload, noise, and flash crowd anomalous scenarios. Also a combination of several anomalies in one observation sequence is investigated as miscellaneous anomalous case showing that SOHMMM is capable of detecting contrasting anomalous cases while HMM-UBM is not.

Table 1.1 demonstrates the proposed methodologies and techniques versus the data sets utilized for experiments and evaluation in each methodology. The FEUP data set is conducted on a real university campus, however do not contain ground truth of anomalous events. The Testbed deployment and wireless simulation data set contain ground truth of a number of anomalous cases generated in them deliberately, however they are reproduced in smaller scales. Although the first data set is rather old, it has lots of data points, small though the second data set is, we can make controlled experiments in a real network, and the third data set is simulated, but allows us to make controlled experiments in relatively large network.

TABLE 1.1: Methodologies vs. Datasets

	Large Data Set - FEUP (Sec. 3.2)	Testbed Deployment (Sec. 3.3)	Wireless Simulation (Sec. 3.4)
HMM Variations (Ch. 4)	✓		
GMM vs. HMM (Sec. 5.2)	✓	✓	
HMM vs. HMM-UBM (Sec. 5.3)			✓
SOHMMM (Ch. 6)			✓

1.5 Research Questions

The most important research questions of the current work, investigated and explored in the presented chapters consist of:

- which model is more efficient in characterizing AP usage in WLAN: single HMM, individual HMMs, or mixture of HMMs. (Chapter 4)
- whether HMMs are required for AP usage time-series analysis and anomaly detection purposes or simpler models like GMMs are adequate. (Chapter 5, Section 5.2)
- whether HMM and HMM-UBM models are capable of anomaly detection and anomalous pattern recognition in AP usage data. (Chapter 5, Section 5.3)
- whether HMM and HMM-UBM models are required for AP anomaly detection or the baseline approaches like RawData and PCA are enough. (Chapter 5, Section 5.3)
- whether HMM-UBM have any advantages over HMM in the context of AP anomaly detection. (Chapter 5, Section 5.3)
- how the spatial connections of various APs in the wireless ground could be considered for model improvement. (Chapter 6)
- whether the integration of SOM and HMM (SOHMMM) has any advantages over simpler model like HMM-UBM or Z-SOHMMM (SOHMMM with zero neighborhood) in the context of AP anomaly detection. (Chapter 6)

1.6 Contributions

This thesis has two main objectives: 1) analysis and modeling of 802.11 AP usage and exploring the inherent relationships between various parts of the network regarding spatial and temporal connectivity, and 2) identification and detection of different types of anomalies and characterizing them efficiently. Both these objectives are investigated on a large data set of AP usage, a smaller Testbed deployment, and a 802.11 wireless network simulation for the purpose of evaluation.

The key contributions of this work include:

- Extracting and presenting two main classes of data features from RADIUS data set: *Density Attributes* containing population related features of user count, session count, and connection duration, in addition to *Usage Attributes* including traffic related features of input and output data in octet, and input and output data in packets. (Chapter 3)
- Deployment of an exploratory Testbed with 1 AP and 6 STAs in FreeRADIUS server, and generating a number of real anomalous cases for experimental purposes. (Chapter 3)
- Conducting 802.11 wireless network simulation in OMNeT++/INET with 10 APs and 100 STAs, and generating several anomalous scenarios for evaluation purposes. (Chapter 3)
- Proposing single HMM (one model for all APs), individual HMMs (one model per AP), and mixture of HMMs (one model per groups of APs) for anomaly detection in 802.11 wireless networks. (Chapter 4)
- Proposing a number of network anomalous patterns deduced from hidden states transition sequences and presenting the potential HMM variations capable of detecting specific types of patterns. (Chapter 4)
- Presentation and comparison of time-invariant modeling approach (GMM) and time-variant modeling approach (HMM) in anomaly detection. (Chapter 5)
- Proposing HMM-UBM as HMMs initialized robustly by Universal Background Model using all the available data. (Chapter 5)
- Application of a hybrid integration of the Self-Organizing Map (SOM) and the Hidden Markov Model (HMM) for the purpose of anomaly detection in 802.11 wireless networks. (Chapter 6)
- Extension of the online gradient descent unsupervised learning algorithm of SOHMMM for multivariate Gaussian emissions. (Chapter 6)

The aforementioned contributions can also be found in the following list of our publications:

- Anisa Allahdadi, Diogo Pernes, Jaime S Cardoso, and Ricardo Morla. "Hidden Markov Models on Self-Organizing Map for Anomaly Detection in 802.11 Wireless Networks". In: *Journal of Engineering Applications of Artificial Intelligence* (2019). Under review
- Anisa Allahdadi and Ricardo Morla. "Anomaly Detection and Modeling in802.11 Wireless Networks". In: *Journal of Network and Systems Management* 27.1 (2019), pp. 3–38. ISSN: 1573-7705. [10]
- Anisa Allahdadi, Ricardo Morla, and Jaime S Cardoso. "802.11 Wireless Simulation and Anomaly Detection using HMM and UBM". In: *arXiv preprint-arXiv:1707.02933*(2017) and under review in *Journal of Simulation SAGE*. [11]
- Anisa Allahdadi, Ricardo Morla, and Jaime S Cardoso. "Outlier detection in 802.11 wireless access points using Hidden Markov Models". In: *Wireless and Mobile Networking Conference (WMNC), 2014 7th IFIP*. IEEE. 2014, pp. 1–8. [12]

- Anisa Allahdadi, Ricardo Morla, Ana Aguiar, and Jaime S Cardoso. "Predicting short 802.11 sessions from RADIUS usagedata". In: *Local Computer Networks Workshops (LCN Workshops), 2013 IEEE 38thConference on*. IEEE. 2013, pp. 1–8. [13]

1.7 Thesis Structure

In chapter 2, we provide a survey of the related work and the most relevant research on usage modeling, anomaly detection in large-scale 802.11 networks, HMM applications in network analysis, integration of HMM and SOM, and wireless network simulation. Chapter 3 deals with the process of data accumulation as a result of wireless users' association attempts in a large data set, a small Testbed deployment, and wireless network simulation. We present the set of main features held in common in these three mentioned data sources, and feature selection techniques for further analysis. In chapter 4 we present various approaches on HMM for anomaly detection and pattern recognition and the experiments conducted on large data set (3.2). The main direction of work in this chapter consists of outlier detection techniques using proposed HMM models including individual HMMs, single HMM, and mixture of HMMs. In chapter 5 we present: 1) time-invariant and time-variant modeling approaches evaluated on large data set (3.2) and Testbed deployment (3.3), as well as 2) improved HMM modeling techniques, namely Universal Background Model (UBM), for detection of anomalous time-series in a database of time-series, distinction of anomalous patterns, and detection of anomalous point within a given time-series. In chapter 6 we introduce a hybrid integration of the Self-Organizing Map (SOM) and the Hidden Markov Model (HMM) for spatio-temporal anomaly detection in AP usage data. We extend the online gradient descent unsupervised learning algorithm of SOHMMM for multivariate Gaussian emissions, and apply the proposed algorithm for the purpose of spatio-temporal anomaly detection in wireless simulation data set (3.4). In chapter 7 we provide major conclusions and reveal potential directions for future work.

Chapter 2

Related Work

2.1 Introduction

In this chapter we analyze existing works on 802.11 usage modeling and anomaly detection, applications of HMM in network analysis, integration of HMM and SOM, and wireless network simulation. These topics particularly comprise the salient lines of research pursued in the upcoming chapters. We begin by analyzing works conducted on the traces of university campuses, corporate organizations, and public 802.11 infrastructures. We present analysis of the recent works related to user behavior in subsection 2.2.1, user mobility modeling at APs in subsection 2.2.2, user encounter patterns in subsection 2.2.3, user access duration in subsection 2.2.4, traffic characterization in subsection 2.2.5, and AP usage characterization in subsection 2.2.6. From AP perspective, all the above mentioned items constitute to the aggregate usage of the 802.11 AP.

Regarding anomaly detection in 802.11 wireless networks, we analyze relevant existing works related to the collected 802.11 usage data, Testbeds, and simulations. The main emphasis is placed on the detection of connectivity and performance anomalies in the usage of 802.11 access points. We focus on the features and techniques used for the detection of AP overload in subsection 2.3.1, AP shutdown/halt in subsection 2.3.2, AP interference in subsection 2.3.3, and wireless measurement tools in subsection 2.3.4.

We also investigate relevant works in the literature concerning the most common applications of hidden Markov models in network analysis. We take into consideration wireless parameters modeling in subsection 2.4.1, user behavior modeling in subsection 2.4.2, prediction of various network features in subsection 2.4.3, network traffic classification in subsection 2.4.4, and eventually anomaly detection in subsection 2.4.5. In each context we provide simple explanation of how HMMs are modeled and employed to resolve specific network issues.

In Section 2.5, we analyze the existing works related to the integration of Self-Organizing Maps (SOM) and Hidden Markov Models (HMM), which is the principal topic of Chapter 6. We study several pieces of research works that take advantage of the synergy of SOM and HMM in collaborating or competing modes. Additionally, we explore resources that refer to the incremental learning process of HMM parameters.

Regarding the network wireless simulation in Section 2.6, we address relevant works that employed wireless simulation for evaluation and validation, obtaining synthesized data and parameterized metrics among others.

2.2 Usage Modeling

Understanding of how WLANs are used provides significant information for those who deploy and manage the network, as well as those who develop systems and applications for wireless networks. Modeling WLAN usage characteristics are beneficial in capacity planning, network monitoring, and providing more accurate network simulation models. Modeling 802.11 AP usage can help performance management of large-scale 802.11 networks, and it is notably important to analyze all aspects of user activities that may aggregate to overall usage at the 802.11 AP.

2.2.1 User behavior

In initial stages of 802.11 wireless network, many researchers focused on understanding how wireless users take advantage of 802.11 infrastructure. In many of these studies, basic statistics about user behavior and network performance were collected and analyzed. For instance in [14] a twelve-week trace of a local-area wireless network of a university building is examined, to explore the overall user behavior, network traffic, and load characteristics. They observed that most users exploit the network for web-surfing, session-oriented and chat-oriented activities. They also found that the entire population can be divided into sub-communities, each with its own unique behavior regarding user mobility, active periods, and traffic generation. Subsequently, the authors analyzed network traces of metropolitan-area wireless network in [15], aiming to understand overall user behavior in a daily basis. They discovered that on average users associate with few APs which are closer together, and the the distance users move can be modeled by Gaussian distribution around the radius of the network. They also found that the most active periods are during the day and evening hours.

Later on, authors in [16] expanded the work conducted in [14] and [15] with broader population using a campus-wide network of 476 APs spread over 161 buildings at Dartmouth College. They identified that residential traffic dominated all other traffic, and web protocols were the largest component of traffic volume. In [17], authors analyzed network traces belong to over 550 APs and 7000 users. They employed several measurement techniques including sys-log messages and SNMP polling among others. They compared the outcome from this trace to a trace taken after the network's initial deployment two years prior in [16]. They found a drastic change in the application usage, with significant increases in peer-to-peer and streaming multimedia traffic. They utilized a new metric for mobility called "session diameter" to show a different mobility characteristics of the embedded devices. They also found that almost half of the users were at the time non-mobile and remain close to home around 98% of the time.

2.2.2 User Mobility

There are numerous efforts in the literature that attempted to characterize the mobility of wireless users among available APs, buildings, and across the entire wireless coverage area. For example, authors in [18] studied user mobility patterns and load distribution across APs. They modeled user mobility with session duration (persistence) and the frequency of users visiting various locations (prevalence). They found that the probability distribution of these two measures follow power laws. They also observed that load is unevenly distributed across APs, some located in popular areas with high number of users, others located in less visited areas and usually idle.

In another related work in [19], authors analyzed daily and weekly periods along with a yearly seasonal effect to understand user mobility and the behavior of APs. They found that a daily pattern is common among users and APs, while a weekly pattern only belongs to APs. Analysis of one year traces revealed the dependency of user mobility and AP popularity on the academic calendar.

Furthermore in [20] authors present an individual mobility-based clustering algorithm that uses roaming events as the metric to evaluate the proximity of APs without considering geographical information. They were able to differentiate network places with social meaning for each individual and APs which are as part of a path between two destinations. In a relative study, authors in [21] explored logical proximity of APs as an alternative measure to physical proximity. They developed an algorithm that uses the ping-pong effect of wireless users between various APs to measure the logical proximity of APs. They provided a logical topology of the APs in a university campus and clustered APs based on their logical distance.

In another direction of work several authors tried to exploit mobility patterns of users to predict their next location or the next AP they will associate with. For example in [22], authors used a two-year trace of the mobility patterns of over 6000 users on Dartmouth's campus-wide Wi-Fi for empirical evaluation of their location predictors. They developed several domain-independent predictors, namely Markov-based, compression-based, PPM, and SPM predictors. Their experimental results showed that low-order Markov predictors outperformed compression-based predictors. However none of their predictors could make a prediction in an unseen data. To overcome this drawback they employed Order-2 Markov predictor with fallback. They also attempted to improve the Markov predictor in terms of space and computation time. In a more recent line of work, authors in [23] analyzed several mobility models for predicting temporal behavior of an individual user. They used fine-grained and continuous mobility data for the evaluation purposes rather than coarse-grained mobility data with partial temporal-coverage in previous researches. They studied the regularity and predictability of human mobility using location-dependent and location-independent models, and showed that a location-dependent predictor is a superior predictor. They found that duration of stay at a location is strongly correlated to the arrival time at the current location and the return-tendency to the next location, rather than recent k location sequences.

The user mobility characterizations of the aforementioned campus WLANs produced almost similar results. The usage is generally diurnal in nature and only a small fraction of devices are mobile. The resulting models mostly investigate distribution of association events from a number of users or a group of them rather than the entire network activity, while we need to explore the whole network activity and usage characteristics.

2.2.3 User Encounter

Several works in the literature tried to extend the analysis of individual users behavior and establish inter-user relationships by observing their association patterns in different locations across 802.11 infrastructures. For example, authors in [24] explored mobile node encounter patterns from campus and enterprise wireless networks using a graph analysis approach. They found that mobile nodes encounter with only a small subset of other nodes (on average between 1.33% to 6.70%), and

the total encounter counts follow the BiPareto distribution. They observed that establishing relationships with only high-ranked friends leads to a disconnected network with separate clusters, while relationships to low-ranked friends provides a reachable network in the encounter-relationship graphs.

Authors in [25, 26] focused on user groups behavior analysis across two university campuses by mining wireless network logs. They used clustering techniques to identify groups of users with similar behavioral patterns. The association patterns of WLAN users symbolized a diverse community with hundreds of distinct behavioral patterns that followed power-law distribution. In another related work in [27], authors identified groups of mobile nodes using two clustering algorithms: k-means chain and spectral clustering. K-means chain identified the number of groups in a dynamic graph, using a chaining process to keep track of group trajectories over the entire trace, while spectral clustering used similarities between node pairs to cluster nodes into groups. They evaluated these algorithms with synthetically-generated traces in addition to a real-world trace from a military scenario. Their results showed that the number of groups and node membership can be accurately extracted from traces, particularly when the number of groups is small.

Moreover, authors in [28] adapted a neural synchrony method [29] to measure the regularity of users visiting to particular locations in weekly basis. They applied their method to three real-world data sets; a metropolitan transport system, a university campus, and an online location-sharing service. Through their experiments they could identify a core group of individuals in each data set that visited at least one location with near-perfect regularity. They also observed a correlation between an individual's most-visited location and irregularity.

2.2.4 User Access to Wireless Network

Other aspects of usage modeling consist of characterizing user wireless access duration that focus on modeling the periods users normally stay connected to APs. For example, authors in [30] analyzed the wireless access patterns, in a university campus, based on mobility, session and visit duration. They showed that the mobility and building type affect the session and visit duration. They also found that mobility and visit duration are in opposite relation, and a family of BiPareto distributions can model the visit and session duration.

In another related work from the [31] authors developed a wireless user model from analysis of five different traces. They defined four behavioral states for the wireless users (active, idle, sleeping, and gone), and measured the transition probabilities between these states. They eventually applied Hidden Markov Models on their data set, and utilized similarity metrics based on HMM likelihood for evaluation purposes. They found similarity in user models across all five traces even though the traces were collected at different venues (library, coffee shops, and conference), and they showed that residing time to APs follow a generalized Pareto distribution.

In the same direction of research, authors in [32] investigated user access time from network simulation. They studied the impact of mobility models on two teletraffic variables: the cell residence time (time connected to an AP) and the handoff rate. They performed various simulations, in OMNeT++ along with INET framework, for different AP layouts and mobility patterns. They observed that the average cell residence time and squared coefficient of variation (SCV) decreased when a memoryless movement pattern is followed (e.g. Random Waypoint) and increased when smoother movement patterns are followed (e.g. Gauss-Markov). Their results

showed that the cell residence time can be characterized by log-normal distributions. They also observed that the Probability Density Function (PDF) of the cell residence time is a combination of a distribution that characterizes very short connections and a distribution that depends on the movement pattern.

We also consider aggregate association duration at APs as one of the main features related to density attributes described in Section 3.2.2.

2.2.5 Traffic Characterization

In addition to user access duration modeling, other works in the literature attempt to focus on infrastructure usage and network performance rather than user behavior. This line of research aims at characterizing traffic of APs and aggregate traffic load of 802.11 infrastructure networks. For example, authors in [33] proposed a time-series forecasting methodology for characterizing traffic at each AP. They conducted their measurement approach in a university campus using the Simple Network Management Protocol (SNMP), and analyzed traffic characteristics in terms of total load and periodicity. They observed spatial locality and diurnal periodicities on heavily utilized APs as well as diurnal periodicities at the total traffic load of the wireless infrastructure.

Authors in [34] proposed a traffic prediction mechanism using the Recursive Least Squares (RLS) algorithm aiming at traffic prediction at short timescales on the order of few minutes. The experimental results of this study showed that the RLS algorithm is capable of accurately predicting the traffic load and shows good adaptive behavior. However, the accuracy of the predictor was constrained by the amount of history required to make a prediction, and how much to predict ahead.

In another related work in [35], authors analyzed measurement traffic statistics for high-speed wireless Internet access sessions collected in a public nation-wide Wi-Fi network. By ranking session lengths and traffic volumes, they found a law implying the truncated Pareto distribution. The basic session traffic variables, for instance session duration download and uploaded traffic volumes are governed by a power law. They also observed that the longer the session the higher the uploaded and downloaded traffic volumes, and the downloaded traffic increases as the uploaded traffic does.

The aggregate traffic load of 802.11 at APs, in terms of number of octet and packets transferred between APs and STAs, form the data features of the current work related to usage attributes, described in Section 3.2.2.

2.2.6 AP Usage Characterization

There exist a number of efforts in the literature that focus more on network characteristics exploring the usage behavior of APs as indicators of different locations around the wireless ground. APs' usage patterns reflect characteristics of spaces where they are deployed, and allow the identification of similarities and differences of those spaces.

Authors in [36] studied the aggregated user workloads both spatially (across multiple locations) and temporally (time of day, and day of week patterns) to understand typical user workload in a large-scale environment in two large cities in the U.S. They explored arrival patterns, arrival models, connection times, and simultaneous users of Wi-Fi APs in different types of venues from small coffee shops to large enterprises. They present a number of modeling techniques to characterize AP usage in terms of arrival counts and temporal variations, connection durations,

and byte counts. The practical use of their models is in capacity planning for sites, building load modules for test purposes, and network monitoring for detection of changes in traffic patterns and intrusion.

Further in [37] authors presented system-wide and AP-level models of traffic demand to capture the network-independent characteristics of the traffic workload. In another work [38] authors conducted analysis on Wi-Fi network as a proxy to space usage modeling aiming to use AP usage data as a means for the characterization of physical spaces, and consequently, as a source of information for a dynamic symbolic model representing those spaces.

In another direction of work [39] AP usage and daily keep-alive events of mobile stations in 802.11 hotspots in infrastructure mode are analyzed and modeled. In this work, generative probabilistic models are investigated such as Gamma mixture of exponentials and Conditional probability models considering dependencies between consecutive samples in time. The generative statistical models and experimental results of this work - conducted on a very similar data set to ours in 3.2 - provided some broad insight into AP usage and illustrated significant aspects of 802.11 wireless networks. In another related work [40] authors analyzed Wi-Fi access point utilization pattern to explore anomalies occur in spaces covered by wireless networks. Their observations consist of 33 week time-series of 230 APs belong to University of Minho Wi-Fi network. They identified four types of APs regarding their usage pattern: normal type, zero type, constant type, and residence type. They proposed statistical anomaly detection techniques to identify spacial events in physical places and feed these events to context-aware applications.

2.3 Anomaly Detection in 802.11 Wireless Networks

In studies concerning 802.11 wireless networks, there exist several analyses on connectivity and performance issues for facilitating the network management tasks. In connection to this, a number of articles investigated overloaded networks, faulty APs, impact of interference in large 802.11 deployments and similar anomalous cases, which are elaborated in the following paragraphs.

2.3.1 Overload Detection

Having explored the network under high-medium utilization conditions, authors in [41] showed that in overloaded networks, stations only maintain a short association period with an AP, and repeated dis-association and re-association attempts are common even in the absence of client mobility. Their analysis demonstrated that stations' throughput suffers drastically from unnecessary hand-offs, leading to suboptimal network performance. In another related work in [42], authors observed that congestion in WLANs leads to the use of lower transmission data rates weakening the overall network throughput and capacity accordingly. To this end, authors presented a technique to measure the utilization of the wireless medium in real-time, and also developed a rate adaptation scheme called Wireless Congestion Optimized Fallback (WOOF). Their experimental results showed that WOOF achieves up to 300% higher throughput in congested networks, compared to other well-known adaptation algorithms.

In another direction of work [43], authors presented a software architecture called DenseAP (DAP), supporting a dense deployment of APs to improve the performance of corporate WLANs. They refrain to apply hardware modifications to the

802.11 standard. In DenseAP architecture, a central controller gathers information from all APs, determines which AP each client should associate with, and also decides on the assignment of channels to APs. The authors demonstrated that DenseAP improves the capacity of an enterprise network by exploiting DAP density via an intelligent association process that encompasses load balancing and dynamic channel allocation.

In a more recent research [44], authors proposed a framework for user association focusing on distributed load balancing in spatially heterogeneous traffic distributions. Their work encompassed various user association policies: rate-optimal, throughput-optimal, delay-optimal, and load-equalizing (α-optimal). They claimed that their α-optimal user association mechanism could achieve a global performance optimum without relying on a centralized controller. They also addressed admission control policies for overloaded situation that blocks all flows at cells edges, while providing a minimum level of connectivity to all spatial locations.

2.3.2 AP Shutdown/Halt Detection

Due to the time varying nature of the wireless medium, it is possible that at any given time a number of APs face problems and stop running. In such circumstances users' connections and throughput will be impaired, leading to significant amount of intermittent sessions, and a busy medium consequently. Automatic detection of unavailable or failed APs is beneficial for effective management of large-scale 802.11 infrastructures. In connection to this, there exist few efforts in the literature that tried to detect halted or crashed APs from 802.11 measurement data. For example authors in [45] presented several algorithms that detect faulty APs by analysis of AP usage logs. They also presented some heuristics to select a path for a technician to repair failed APs. The main assumption in their algorithm is that the longer the time an AP does not register events, the greater the probability that particular AP is crashed or halted. Their evaluation results showed that their best algorithm can detect up to 90% of failed APs by processing log files at Dartmouth College. Authors in [46] made use of clock skew of APs as their fingerprint to detect unauthorized APs. They calculated the clock skew of an AP from the IEEE 802.11 Time Synchronization Function (TSF) time stamps sent out in the beacon/probe response frames. They applied two methods- linear programming and least-square fit- along with a heuristic for differentiating original packets from those sent by the fake APs. Their experimental results showed that clock skews remain consistent over time for the same AP but vary significantly across APs.

In another direction of work, authors in [47] proposed and evaluated a technique called Client Conduit, which enables bootstrapping and fault diagnosis of disconnected clients. They utilized a controller system along with a diagnostic server to detect and self-diagnose disconnected clients around the faulty APs. In their solution, clients are augmented to begin an ad-hoc network when an AP stops functioning.

The focus in these works is in supporting disconnected clients not to associate with a crashed AP, but for that clients require cooperation among themselves and support from a third party device such as diagnostic server or management station. Instrumenting devices (APs and clients) and presentation of third party devices in large-scale 802.11 networks can be challenging and expensive due to high population of users and APs.

2.3.3 Interference Detection

Interference in 802.11 networks often occurs where transmission in one link of the network interferes with the transmissions in other neighboring links. It can also be induced by other radio waves in the same frequency range. During the interference period many re-transmissions are required that results in overall 802.11 performance degradation. There exists several researches in the literature that investigate the effects of interference and propose techniques to detect and mitigate its potential impact.

For example, authors in [48] studied the impact of interference in chaotic 802.11 deployments on end-client performance. Having used large-scale measurement data, they showed that it is not uncommon to have tens of APs deployed in close proximity, and most APs are not configured to minimize interference with their neighbors. They designed and evaluated automated power control and rate adaptation algorithms to minimize interference among neighboring APs to ensure robust end-client performance.

In another related work [49], authors proposed methodologies including intelligent frequency allocation across APs, load balancing of user affiliations across APs and AP adaptive power control for interference mitigation in dense 802.11 deployments. Furthermore, authors in [50] studied the impact of RF interference on 802.11 networks from devices like Zigbee and cordless phones that crowd the 2.4GHz ISM band to devices like wireless camera jammers and non-compliant 802.11 devices that disrupt 802.11 operations. They affirmed through practice that moving to a different channel is more effective in coping with interference than changing 802.11 operational parameters such as CCA (clear channel assessment).

In [51], a usage pattern called "abrupt ending" is explored in a data set similar to 3.2 that concerns the dis-association of a large number of wireless sessions in the same AP within a one second window, or in a nutshell "simultaneous session ending". The authors introduced some anomalous patterns that might be in correlation with the occurrence of this phenomena. For instance, AP halt/crash, AP overload, persistence interference and intermittent connectivity. The analysis of the anomaly-related patterns performed in this research, inspired our work to regenerate similar anomalies via network simulation (3.4) in addition to the real Testbed deployment (3.3).

2.3.4 Wireless Measurement Tools

Several prior works are dedicated to studying the dynamics of wireless network behavior, as well as the performance and reliability of WLAN technologies [8, 52, 53, 54]. In [8] a system called Jigsaw is presented which uses multiple monitors to provide a single unified view of physical, link, network and transport-layer activities, including inference techniques for the particular issues of 802.11. The authors deployed an infrastructure with over 150 radio monitors that capture 802.11b and 802.11g activities in a university building to investigate the causes of performance degradation. Significant challenges of such vast distributed monitoring system include the necessity of hardware and software instrumentations on each and every monitor and the scalable synchronization difficulties and inaccuracies. For this reason, most wireless management techniques avoid broad modifications in the clients devices, sensors, sniffers and monitors deployed in the large wireless covered area.

In another line of research a Passive Interference Estimator (PIE) is presented in [52] which provides a fine-grained estimation of link interferences in WLAN. PIE

provides an estimate of WLAN interference caused by client mobility, dynamic traffic loads, and varying channel conditions. This work is inspired by two previous WLAN monitoring approaches: the aforementioned Jigsaw [8] and WIT [55]. The PIE producers used sniffing at APs to avoid deploying additional monitors similar to Jigsaw, but with the penalty of missing a portion of uplink client traffics and hence uplink client conflicts. However, they proposed an accurate approach in estimating link interference by providing a conflict graph in real time.

In a similar direction of work, fine-grained detection algorithms are proposed that are able to distinguish the root-causes of performance degradation at the physical layer [53]. It is described that various faults, such as hidden terminals, capture effects and noise, could have the same propagation effects on the network layer (degraded throughput) and therefore could lead to the same remediation techniques from 802.11 (rate fallback), while they have completely different origins in the physical layer. Hence, the researchers of this work designed a unified framework for this purpose, called MOJO, that combines the observations from multiple distributed sniffers and diagnoses the granularity of the root causes to suggest appropriate remedies for different physical faults. Although the proposed framework measures the impact of the most commonly observed faults on different network layers, it is still a client side monitoring system and suffers from the extensive sniffer distribution all over the wireless covered area.

WiMed [54] uses only local measurements from commodity 802.11 NICs for understanding how the medium is utilized, and for inspecting the causes of interferences (including non-802.11 devices). WiMed provides a time-domain view of how the medium is used in a given 802.11 channel, and identifies the root causes of interference using physical layer properties such as bit error patterns and medium busy times. The authors refrained from elaborated instrumentation and dedicated infrastructure, however detectors are only implemented for interference and contention, and there is a higher confidence for recognition of non-802.11 interferer rather than 802.11 sources of interference.

The mentioned studies of this subsection expose the difficulties in monitoring the wireless environment thoroughly, and the challenges of performance estimation in these complex networks. Most cases - require heavy instrumentation of the user devices and focus on specific anomalies affecting individual users - thus neither considering usage trend nor location related anomalies.

2.4 HMM Applications in Network Analysis

In wireless networking, HMMs are employed to address various aspects of network measurement and analysis.

2.4.1 Wireless Parameters Modeling

Hierarchical and Hidden Markov based techniques are analyzed in [56] to model 802.11b MAC-to-MAC channel behavior in terms of bit error and packet loss. The authors employed two random variables in packet loss process, inter-arrival-rate and burst-length of packet loss, and applied the traditional two-state Markov chain. The results demonstrates that two-state Markov chain provides an adequate model for the 802.11b MAC-to-MAC packet loss process. Furthermore, in regard to bit error modeling, three other Markov-based chains are evaluated: full-state, hidden,

and hierarchical Markov chains. It is illustrated that among these chains, the full-state Markov bit error model of order 9 and above yields the best performance. Since the main concern to use HMM in this example is to generate error traces, a simple three-state HMM is designed and utilized for one HMM solution: the adjustment of model parameters to best account for the observed signal.

In a more recent line of research in [57] a multilevel approach involving HMMs and Mixtures of Multivariate Bernoullis (MMB) is proposed to model the long and short time scale behavior of wireless sensor network links, that is, the binary sequence or trace of packet receptions (1s) and losses (0s) in the link. In this approach, HMM is applied to model the long-term evolution of the trace, and the hidden states correspond to packet reception rate. Within the aforementioned hidden states, the short-term evolution of the trace is modeled by either another HMM or by a MMB. That is how the multilevel, or in this case the two level approach, is formed. The notion of multilevel HMM is an impressive concept regarding anomalous pattern detection in our work.

Authors in [58] applied HMMs for spectrum sensing in cognitive radios, since the true states (occupancy by primary users) of a sub-band (idle frequency) are never known (hidden) to the cognitive radio. They employed an HMM to model the evolution of occupancy/non-occupancy of a sub-band by its primary user over time using the measurements obtained by the cognitive radio. The hidden sequence is considered to be the sub-band occupancy state, and the observed sequence is expressed as the sequence of decisions generated by the sensing technique of a second user. The authors utilized maximum likelihood approach and the Viterbi algorithm to estimate the hidden states. They validated the accuracy of their proposed method in predicting the true states of a sub-band using extensive simulations.

2.4.2 User Behavior Modeling

In a very recent line of research in [59] authors employed patterns of data traffic to identify the function of physical locations. They investigated the predictability of mobile app usage behavior and mobility behavior of users by observing users' mobility behavior/app usage behavior. They built a hidden Markov model (HMM)-based predictor to characterize a user's mobility/app usage behavior. They showed that there is a strong correlation between users' app usage behavior, mobility behavior, and data traffic patterns.

In another direction of work in [60], authors proposed an analytical method to classify web user behavior based on latent states of users as intention, interest, or motivation, and applied their method to the data of a social network game. They put the click-stream data of many users into a Hidden Markov Model in which the number of hidden states is large enough to build a state transition network. They represented the movement on the state transition network as user behavior, and found their approach suitable for services with many pages and complicated to analyze.

Further in [61], authors investigated heterogeneous cellular networks and the hand-over strategy by analyzing the self-similar least-action human walk (SLAW) and proposing a method based on the hidden Markov model for perceiving user behavior in hot spots. They simulated users' mobile paths in hot spots based on SLAW, and modeled user behaviors using HMM. Then, they predicted the corresponding moving time by the mobile sequence of the user for designing a handover management plan.

2.4.3 Network Parameter Prediction

One of the salient application of HMMs addressed in wireless networking is network parameter prediction. In a recent research, HMMs are utilized to model and predict the spectrum occupancy of sharing radio bands [62]. The channel status prediction is considered as a binary series prediction problem, as channel occupancy can be represented as idle or busy depending on the presence or absence of a primary user activity. An ergodic two-state discrete HMM is employed to this problem. The experimental results showed the ability of HMM in offering a new paradigm for predicting channel behavior in cognitive radio. Some other prominent work has been done on a similar subject in radio spectrum sensing and status prediction using HMMs [63, 64, 65]. For example, authors in [65] designed a channel status predictor using two different adaptive schemes: a neural network based on multilayer perceptron (MLP), and a hidden Markov model (HMM). The simulation results demonstrated that the two prediction schemes performed similarly under the same traffic scenario while the performance of MLP predictor is slightly better than that of the HMM predictor. However the HMM predictor is more appropriate for time varying traffic scenarios while the MLP predictor should be retrained periodically for better performance.

Furthermore, in another direction of work, HMMs are applied for modeling and prediction of user movement in wireless networks to address issues in Quality of Service (QoS) [66]. User movement from an AP to an adjacent AP is modeled using a second-order HMM. Although the authors demonstrated the necessity of using HMM instead of Markov chain model, the proposed model is only practical for small wireless networks with a few number of APs, not huge enterprises or widespread networks. In [67] authors proposed a new approach for optimizing the hand-off decision in Femtocell networks using Hidden Markov Model. They formulated the hand-off problem as an optimization problem whose objective is to find the best Femtocell Access Point (FAP) assignment strategy that minimizes the number of unnecessary hand-offs while maintaining a good quality of wireless communications. The HMM is employed to predict the target FAP by observing the geographic positions of the mobile user. The simulation results showed that the proposed approach minimized the number of hand-offs and enhanced the dwell time in the FAP in comparison with others hand-off decision making strategies.

2.4.4 Network Traffic Classification

Network traffic classification is the process of analyzing traffic flows and associating them to peculiar categories of network applications. Traffic modeling and classification gained significance in many areas such as bandwidth management, traffic analysis, prediction and engineering, network planning, Quality of Service provisioning and anomalous traffic detection.

There are various efforts in the literature that tried to exploit HMMs for network traffic classification. For example, authors in [68, 69] proposed a packet-level traffic classification approach based on Hidden Markov Model (HMM) using real network traffic and estimating Packet Size (PS) and Inter Packet Time (IPT) characteristics. They considered their HMM model with discrete states and continuous bi-dimensional observations consist of IPT and PS measurements. In [68] the proposed HMM model is applied to classify real traffic traces of a network game (AoM), SMTP, and HTTP. Moreover the model is used for monitoring to obtain an estimate of the current state via the Viterbi algorithm, and prediction on the basis of the current state estimate and of the trained model parameters. In [69] the classification is

extended to broader range of applications: HTTP, SMTP, Edonkey, PPlive, and MSN Messenger among others. The presented results demonstrated that the proposed Packet-Level Hidden Markov Model (PL-HMM) could be a good candidate as to be used in a multi-classification scenario (when different classification engines are used and their output is combined by a decision system).

In another related work, authors in [70] investigated the performance of a new Hidden Markov Model structure used as the core of an Internet traffic classifier and compares the results against other models presented in the literature. The HMM structure includes the packet payload size (PS) and the inter-packet times (IPT) sequences, and consist of an HMM profile associated with a fully-connected HMM. The first part (HMM profile) captures the specific properties of the initial protocol packets while the second part (fully connected HMM) captures the statistical properties of the entire sequence present in the flow. The results showed the superior performance of the proposed mixed model (5-Profile+5-Fully HMM) against five other models used in the literature including HMM Profile (5P-HMM), fully connected HMM (5F-HMM), the centroid, 1-NN and Naive Bayes classifiers.

2.4.5 Anomaly Detection

Authors in [71, 72] attempted to estimate the interference between nodes and links in a live wireless network by deploying several sniffers across the network to capture wireless traffic traces in a passive mode. They modeled the 802.11 MAC as a Hidden Markov Model, and learned the state transition probabilities in this model using the observed trace. The HMM approach is used for modeling interactions between a pair of senders in an 802.11 network and inferring sender-side interference relations (deferral behavior). A sender node is considered to be in one of four states: idle, backoff, defer, and transmit, and the interference is defined as transition into defer state. They assumed that a single Markov model is not enough to address the complete behavior of the network, so they formed a combined Markov model with each state consisting of 2-tuple states of individual nodes. Therefore their model is restricted to determine pairwise interference relationships. The observation symbols contain the status of the pair of nodes (e.g. one, both or none transmitting). Their Experimental results demonstrated that the HMM approach is more accurate than simpler heuristics.

There are several attempts in the literature that investigated intrusion detection techniques using HMMs. For example, authors in [73] explored an HMM strategy for intrusion detection using a multivariate Gaussian model for observations that are used to predict an attack that exists in a form of a hidden state. Their proposed model contains a self-organizing network for event clustering, an observation classifier, a drift detector, a profile estimator, a Gaussian mixture model (GMM) accelerator, and an HMM engine. The hidden states are defined as follows: normal, hostile intrusion attempt, friendly intrusion attempt, intrusion in progress, and intrusion successful. They observed that the intrusion state cannot be inferred directly by monitoring any specific parameters, hence they intended to predict an attack based on mixture of observable data-points, events and current states. Accordingly they designed a statistical mechanism for intrusion prediction using HMM with weighted Gaussian mixtures as observed data. They used the proposed methodology to predict the intrusion states based on observation deviation from normal profiles or by fitting it into an appropriate attack profile. In a more recent research in [74] authors defined a sequence of attack states corresponding to the attack stages and the proposed detection system adopts a Hidden Markov Model for detecting advanced

planned intrusion attacks. The proposed HMM based classification model consists of three layers: the hidden states at the first layer, the observable events emitted from the hidden states at the second layer, and the feature set used for correlation at the third layer, and different attack events might refer to different features. The experimental results showed that the proposed detection system can identify the attacks efficiently.

In a different direction of work [75] authors investigated anomaly detection of human dynamics using spatio-temporal data obtained from GPS facilities. They formalized their problem as a semi-supervised anomaly detection problem that detects contextual anomalies behind time-series data. Their anomaly detection method is based on a sticky hierarchical Dirichlet process hidden Markov model, which is able to estimate the number of latent states according to the input data. They obtained significant detection results through experiments in synthetic data as well as real gridded population data, in which anomalies were detected when and where an actual social event had occurred.

To the best of our knowledge, HMM related studies in wireless network management are rarely employed in performance prediction and anomaly detection.

2.5 Integration of HMM and SOM

In this section, we analyze the existing work related to the integration of Self-Organizing Maps (SOM) and Hidden Markov Models (HMM), which is the principal topic of Chapter 6. There are a number of attempts in the literature that tried to integrate the SOM and the HMM in different ways [76, 77, 78, 79, 80]. In [76] Spherical Self Organizing Map (S-SOM) is proposed, which uses HMM models as neurons (S-HMM-SOM) for the purpose of classifying the time series data. The HMM models in [76] are discrete and the author applied Baum-Welch algorithm for updating the model parameters. In [77] the authors extended the self-organizing mixture models for multivariate time-series assuming that the time-series are generated by HMMs. This model which is called self-organizing hidden Markov model (SOHMM), uses constrained Expectation Maximization (EM) for HMM parameters estimation. The self-organization in this work is used for the purpose of meteorological states visualization.

In another direction of work in [78], a self-organizing Markov map (SOMM)-based architecture is presented for hand gesture recognition. The approach involves a combination of SOMs and Markov models for gesture trajectory classification. In this work the neurons on the SOM map correspond to the states of the Markov models. In [79] the combination of SOM and HMM (SOS-HMM: Self Organizing Structure of HMM) automatically extracts the structure of an HMM without any prior knowledge of the application domain. In this model the macro-HMM is represented as a graph of macro-states, where each state represents a micro-HMM. In conclusion, each neurons in SOS-HMM collaborative architecture is either HMM by itself or HMM hidden state. In our work each particular neuron on SOM lattice is associated with an HMM.

In [80] the fusion and synergy of SOM and HMM are employed in biological molecules studies to meet the increasing requirements imposed by the properties of deoxyribonucleic acid (DNA), ribonucleic acid (RNA), and protein chain molecules.

They proposed a stochastic unsupervised learning algorithm based on the integration of the SOM and the HMM principles, called SOHMMM. The authors demonstrated the SOHMMM's characteristics and capabilities through two series of experiments based on artificial sequence data and splice junction gene sequences. However, in [80] only the discrete observation setting is addressed. In this thesis we improve this algorithm for multivariate Gaussian emissions, and extend the model to fit the requirements of our anomaly detection study.

The training process of SOM and HMM sub-units are in most cases disjoint and conducted independently. However, there are two main approaches regarding this hybrid technique. First approach considers SOM as a front-end processor (e.g. vector quantization, preprocessing, feature extraction) and HMMs are then used in higher processing stages [81, 82]. The second approach places the SOM on top of the HMM [80, 83].

Incremental learning of HMM parameters is the core function of the SOHMMM algorithm which is based on a stochastic gradient descent technique. Incremental learning of new data sequences allows to adapt HMM parameters as new data becomes available, without having to retrain from the start on all accumulated training data. There are various techniques in the literature that address this topic. These techniques are classified according to the objective function, optimization technique and target application, involving block-wise and symbol-wise learning of parameters. The authors in [84] presented a comprehensive survey of techniques that are suitable for incremental learning of HMM parameters, among which the stochastic gradient descent technique of SOHMMM is referred as one of the numerical optimization methods.

Additionally, there exist few efforts in the literature that exploit the SOM and the HMM for anomaly detection purposes [85, 86]. In [85] the authors presented an intrusion detection system in which the SOM determines the optimal measures of audit data and reduces them into appropriate size for efficient modeling by HMM. Similar to our previous work [12] two types of HMMs are utilized: single model for all the users and individual models for each user. In another relevant work in [86], the HMM and the SOM are investigated separately as intrusion detection techniques. Testing results show that the HMM method using the transition property of events outperformed SOM using the frequency property of events. Regarding the same subject of intrusion detection, SOM and HMM have a collaborating connection in [85] and competitive roles in [86].

In Chapter 6 we intend to benefit from the collaboration of these two techniques (SOM and HMM) to extend our anomaly detection framework applying only HMM [12, 10, 11].

2.6 Wireless Network Simulation

There are numerous efforts in the literature that try to exploit simulation as an effective tool to setup a computationally tractable network. Wireless network simulation is used for various objectives from assessment and validation of models to obtain synthesized data and parameterized metrics.

In [37] the authors employed simulation to generate synthetic traffic and validate their proposed model of traffic workload in a campus WLAN. As another example, researchers proposed a framework in [87] to integrate the infrastructure mode and ad hoc mode in WLANs and they implemented the framework in NS2 [88]. They used simulation to show the higher performance of their proposed model compared

to the traditional wireless LAN. In a rather relevant work to ours, the performance of IEEE 802.11 wireless networks is evaluated using OPNET Modeler in [89]. The simulated network in infrastructure mode for one AP and 12 stations investigated the performance of pure 802.11g network over a network that uses both 802.11g and 802.11b clients.

After exploring various simulation frameworks such as NS3 [90] and OPNET [91], we chose to use OMNeT++ [92] along with INET Framework [93] for our research purposes. Regarding OMNet++, and its simulation models, a number of articles worked on validating the reliability and accuracy of OMNeT++. For example in [94] the authors performed a measurement study of wireless networks in a highly controlled environment to validate the IEEE 8021.11g model of OMNet++. They used metrics like throughput, delay and packet inter-transmission to compare the measurement results to identical simulations. They showed that the simulation results match the measurements well in most cases. Furthermore in [95] the reliability of OMNeT++ is assessed for wireless DoS attacks by comparing the simulation results to the real 802.11 Testbed. In this case, throughput, end-to-end delay, and packet lost ratio are considered as performance measures. The authors confirmed the accuracy of the simulation results in wireless DoS domain.

However, there exist few efforts in the literature that conducted simulation of WLANs in OMNeT++ concerning performance and quality of service (QoS). For example in [96] the performance of the TCP protocol for audio and video transmission is evaluated using OMNeT++ simulation. In another direction of work in [97] an overview of the IEEE 802.11b model is simulated in OMNeT++ and an example network consisting of a mobile station moving through a series of APs is used to analyze the handover behavior of the model. To the best of our knowledge the simulation of aforementioned anomalous patterns in WLAN infrastructure mode has never been done before.

Chapter 3

Experimental Setup

3.1 Introduction

In this chapter we address three main data sets we utilize for experimental analysis and evaluation through the rest of the thesis. The schemas of all these three data sets are identical and related to the wireless station RADIUS authentication data collected at access points. However, each set of data has different sources. The first data set explained in section 3.2 belongs to the RADIUS authentication log data collected at the hotspot of the Faculty of Engineering of the University of Porto (FEUP). It is a large data set as it contains the connection summary of more than 45 thousand users associated to 364 APs. The second data set described in section 3.3 is related to an exploratory small Testbed deployed at a home environment with 6 wireless users connected to 1 AP. The Testbed uses the FreeRADIUS server and a set of well defined anomalous cases are generated in this network for the evaluation purposes. The last data set is described in section 3.4 and is generated through wireless network simulation using the OMNeT++ [92] wireless simulator and INET framework [93]. The simulated wireless network consist of 100 wireless users associated to 10 APs located in a $1500m \times 1200m$ wireless ground. A number of anomalous scenarios are generated that mostly affect physical layer parameters. The generated anomalous cases in Testbed and wireless simulation experiments are not exactly the same, but there are both employed for evaluating similar types of anomalies. Although the first data set is rather old, it has lots of data points, small though the second data set is, we can make controlled experiments in a real network, and the third data set is simulated, but allows us to make controlled experiments in relatively large network.

3.2 Large Data Set

As a first set of data, we make use of RADIUS authentication log data collected at the hotspot of the Faculty of Engineering of the University of Porto (FEUP). The University hotspots are part of the Eduroam European wireless academic network initiative. The entire data set incorporates records of 802.11 mobile stations' association to APs stored at a RADIUS authentication server. When a client associates/disassociates to an 802.11 AP, a "START"/"STOP" event is recorded. A probing log event "ALIVE" is generated every 10 or 15 minutes (depends on the server configuration) while the client is still connected to the network [3]. Each log record includes some key attributes of time-stamp, session ID, association duration, number of input and output packets/octets. Table 3.1 presents a brief explanation of some of these key attributes more relevant to this work.

TABLE 3.1: The key attributes of RADIUS accounting table

Acct-Status-Type	has three values: Start, Alive and Stop. A Start record is created when a user session begins. An Alive record is registered after each 10 or 15 minutes for the users that are still connected. A Stop record is generated when the session ends.
Acct-Session-Id	is a unique number assigned to each session to facilitate matching the Start and Stop records in a detail file, and to eliminate duplicate records.
Acct-Session-Time	records the user's connection time in seconds. This information could be included in Alive or Stop records.
Acct-Delay-Time	is the number of seconds passed between the event and the current attempt to send the record. The approximate time of an event can be determined by subtracting the Acct-Delay-Time from the time of the record's arrival on the RADIUS accounting server.
Called-Station-Id & Calling-Station-Id	record the IP address of the AP (Called Station) and the wireless user (Calling Station) connected to that AP.
Timestamp	records the time of arrival on the RADIUS Accounting host measured in seconds since the epoch (00:00 January 1, 1970). It provides a machine-friendly version of the logging time at the beginning of the accounting record.
Acct-Input-Octets & Acct-Output-Octets	records the number of bytes received (Acct-Input-Octets) and sent (Acct-Output-Octets) during a session. These values appear in Alive or Stop records.
Acct-Input-Packets & Acct-Output-Packets	records the number of packets received (Acct-Input-Packets) and sent (Acct-Output-Packets) during a session. These values appear in Alive or Stop records.

Our trace data consists of the daily summary of connections between 364 APs and their corresponding wireless stations collected in almost two years, from January 1, 2010 to December 22, 2011. The university campus contains over 30 buildings, including classrooms, administrative offices, auditoriums, libraries, cafeterias, laboratories, etc. During the mentioned period, the usage record of more than 45 thousand users was observed through the established connections of over 24 million sessions. Table 3.2 depicts the evolution of the usage across the hotspot throughout the academic semesters.

TABLE 3.2: The semester-level evolution of hotspot usage during two years

Academic Semesters	# APs	# Users	# Sessions	Total Input Traffic (TB)	Total Output Traffic (TB)
Spring 10/11	238	15564	5127823	148	253
Fall 10/11	278	15614	2619497	81	138
Spring 11/12	317	20200	5879742	177	359
Fall 11/12	338	21946	7167023	91	170

In general, an increasing trend is observed in the number of deployed APs, number of wireless users and overall number of RADIUS sessions (start, alive, and stop), from semester to semester. Total input and output traffic, however, fluctuate between spring and fall semesters to some extent. Although the overall sent and received traffic grows in volume in ultimate fall/spring semester rather than the earlier, the wireless network are subjected to higher traffic in spring semesters compared to fall semesters.

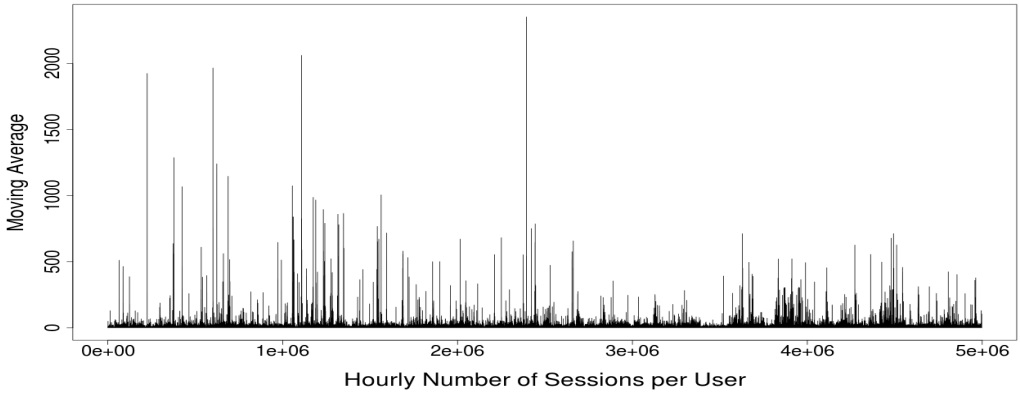


FIGURE 3.1: Moving average of the hourly number of sessions per user.

3.2.1 Preliminary Data Analysis

In this section we present some extensive statistical analysis about the entire data set and demonstrate relevant graphics revealing some general facts of underlying usage pattern of FEUP wireless network. We conduct this study from two peculiar viewpoints: 1) users and their sessions, and 2) access points and their users.

User Sessions

As indicated earlier, each user can connect to the same AP more than once during a day, and each connection creates a separate sessionID in the accounting table. An ideal association to the wireless network could last for the entire day and if the user is fixed in its location, it is expected to have the same session without interruption. However, this is not always the case and users disassociate from their current AP and associated to the same AP or another AP in the vicinity for various reasons such as signal strength loss, and roaming among other reasons.

Figure 3.1 shows the moving average of the number of sessions per user during one hour of connection. Although the majority of users have a few number of sessions in an hour which shows few number of dis-associations, extreme cases are also detectable in this figure. For instance, users are observed that generate over 2000 sessions on average in an hourly connection to a single AP. To study the greatest population of users, Cumulative Distribution Function (CDF) of users and their containing sessions is demonstrated in Figure 3.2. This figure displays that more than 70% of user connections remain unbroken and preserve a single session during the hourly association to their affiliated AP, and over 95% of user connections contain only 5 sessions during an hour which is the result of intentional or unintentional disassociation from the current AP.

Figure 3.3 encloses similar information as Figure 3.1, but in a daily basis. As expected, the number of dis-associations during one day is higher than an hour period. Figure 3.4 demonstrates that users holding a single session during a day, are less than 40%. Such connections could be issued from stationary or idle users in vacant locations of the campus with few or none other active users around, or the user could be connected for a short time containing only one session. This figure also displays that about 20% of the sessions are interrupted between 5 and 20 times a day.

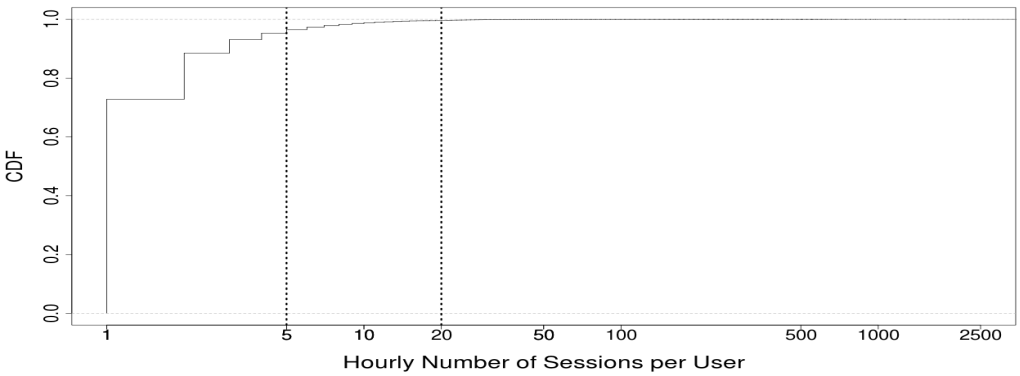


FIGURE 3.2: CDF of the hourly number of sessions per user.

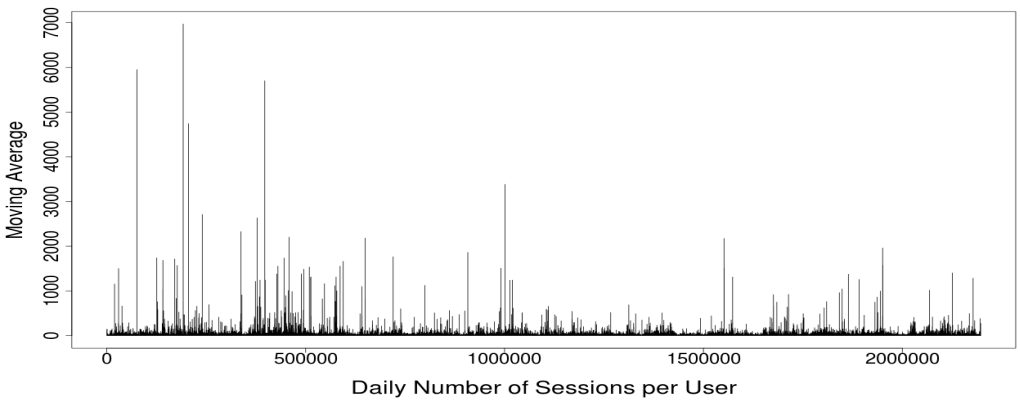


FIGURE 3.3: Moving average of the daily number of sessions per user.

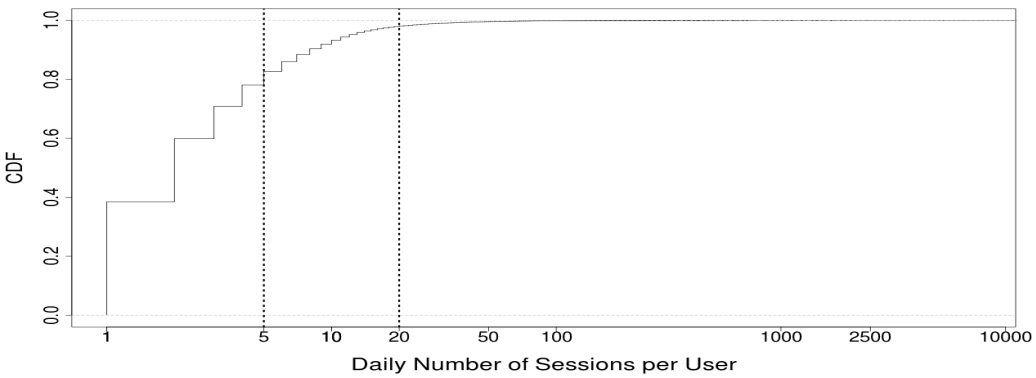


FIGURE 3.4: CDF of the daily number of sessions per user.

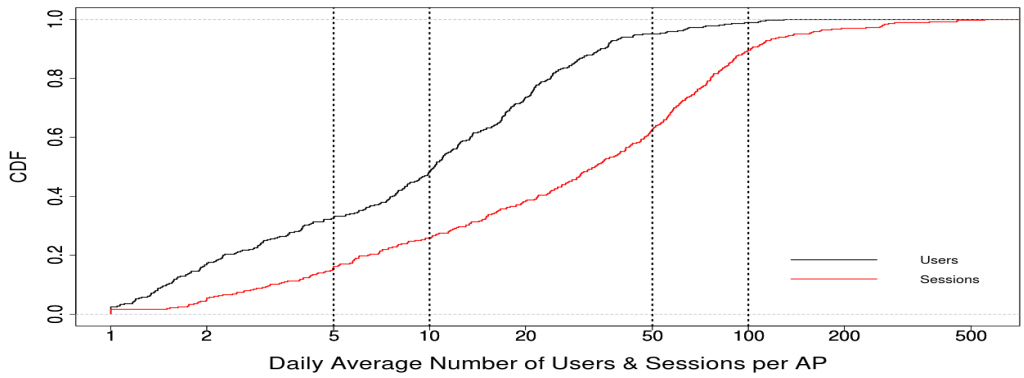


FIGURE 3.5: CDF of average number of users & sessions per AP.

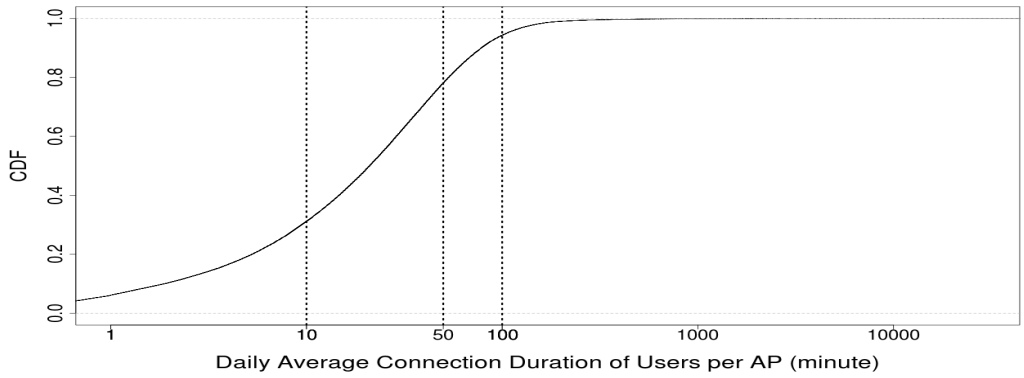


FIGURE 3.6: CDF of the daily average connection duration of users per AP (minute).

Access Points

In this part, we focus on the usage behavior of APs as indicators of different locations around the university campus. Figure 3.5 demonstrates the average number of users and sessions per AP during the two years of experiment for the working days only. Because usage in the campus on weekends is substantially reduced, the weekends are excluded from this statistics. These statistics could differ from semester to semester as the number of users and their corresponding sessions evolve over time, however this figure provides a general report of involvement of the entire set of APs in the wireless covered area. Figure 3.5 shows that around half of the APs associate with 10 users during a day. This figure also indicates that few APs (only 5%) have a large number of users per day (more than 50), and about 30% of the APs typically associate with only 5 users each day. This figure also demonstrates that few APs (almost 10%) contain more than 100 sessions during a daily connection which are generated by less than 40 users.

Figure 3.6 reveals interesting information about the duration of users' daily connections per AP. It shows that the average connection period of users in 30% of the APs is only 10 minutes per day. This data most probably belongs to the mobile users, guests or short-term users. Figure 3.6 also demonstrates that in about 95% of the APs, users maintain their connections at most for 100 minutes (less than 2 hours) a day.

The information provided in Figure 3.5 and 3.6 - can be related to the importance of the APs and their locations, for instance whether they are located in a busy entrance hall or a quiet corner of the campus. Such sort of information also implies potential categories in terms of university divisions like administrative office, classroom, cafeteria, auditorium, etc. Such classification plays an important role for further analysis and modeling practices and brings about the question of the extent of similarity or difference of usage patterns in similar categories with different population of users.

3.2.2 Data Features

A number of features emerge from the raw data set as a result of a preliminary analysis and enumeration process on a timely basis of 15 minutes. We categorize all the measured features as two main classes: *Density Attributes* and *Usage Attributes*. Those features that are indicators of density, basically demonstrate how crowded is the place in terms of current users. The usage features disclose the amount of sent and received traffic by the current users. The former attributes mainly characterize the association population and durability, while the later attributes reveal the total network traffic regardless of how populous the place is.

Density Attributes

User Count: the number of unique users observed in a specific location (indicated by an AP) during the predefined time-slot (15 min).

Session Count: the total population of active sessions during a time-slot regardless of the user. This attribute reveals the number of attempts made by all users to associate with the current AP. The connection duration of each user consists of one to many sessions.

Connection Duration: the total duration of association time of all the current users of an AP in a 15 min time slot. This attribute is an indicator of the overall connection persistence. The maximum amount of this features is achieved when there is no evidence of disassociation in the ongoing active sessions during a time slot ($User\ Count * 15\,min$).

Usage Attributes

Input Data in Octets: the number of octets transmitted from the client and incoming to the NAS port, and is only present in the Stop or Alive sessions. This attribute briefly refers to the number of bytes uploaded by the wireless user.

Output Data in Octets: similar to Input Data in Octets but receives by the client rather than sent, and it refers to the number of bytes downloaded by the wireless user.

Input Data in Packets: similar to Input Data in Octets, just to be measured in packets instead of bytes.

Output Data in packets: similar to the above Output Data in Octets, just to be measured in packets instead of bytes.

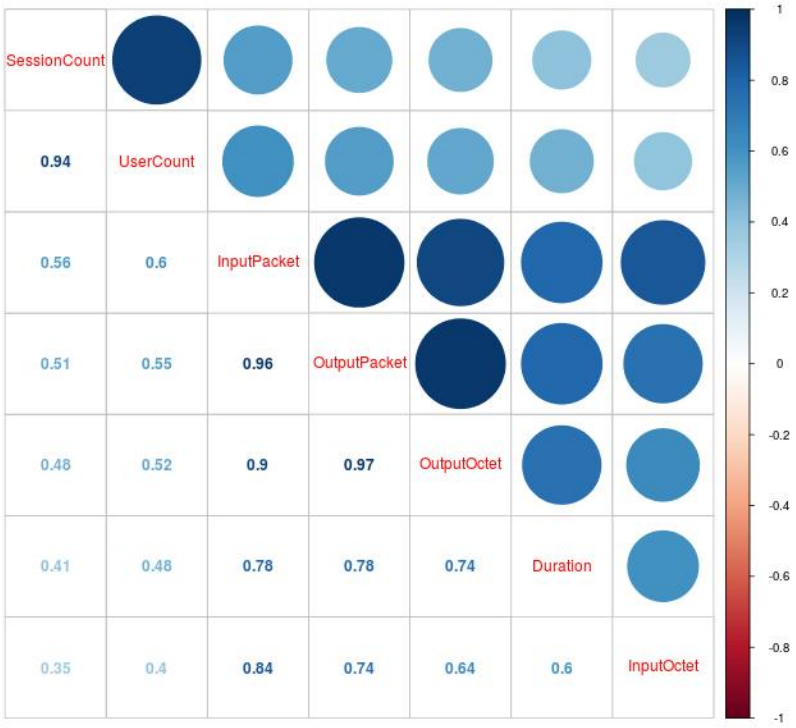


FIGURE 3.7: Correlation matrix of the main data features - features stored in no particular order.

3.2.3 Analysis of Features

In this section we discuss the correlation of the data features explained earlier in an attempt to select an appropriate set of features for further analysis. Figure 3.7 depicts the correlation matrix of all the above features. There is a high correlation observed between *User Count* and *Session Count* (0.94), on the grounds that the number of sessions are always equal or higher than the number of users in a time-slot. *Duration* do not have a strong correlation with any of the mentioned features, neither with *Density Attributes*, nor with *Usage Attributes*.

Having considered the input and output traffic transferred in octets, there is no significant correlation between these two (0.64) compared to Input and output data in packets (0.96). However there is a noticeable correlation between *Output Octets* and its corresponding attribute *Output Packets* (0.97), as well as *Input Octets* and *Input Packets* (0.84). Nevertheless we consider input/output data in octets and in packets as semi-independent variables, and include both of them in our further experiments. The information added to the system through input and output traffic in octets simply take into account all the sent and received data in bytes. However, the input and output traffic measured in packets could bring other types of information as the packets' size could differ by various factors such as application types and communication protocols.

For subsequent analysis and modeling procedures, we favor using less features rather than the entire set of attributes introduced earlier. For this purpose, we applied Principal Component Analysis (PCA) technique to find the combination of the

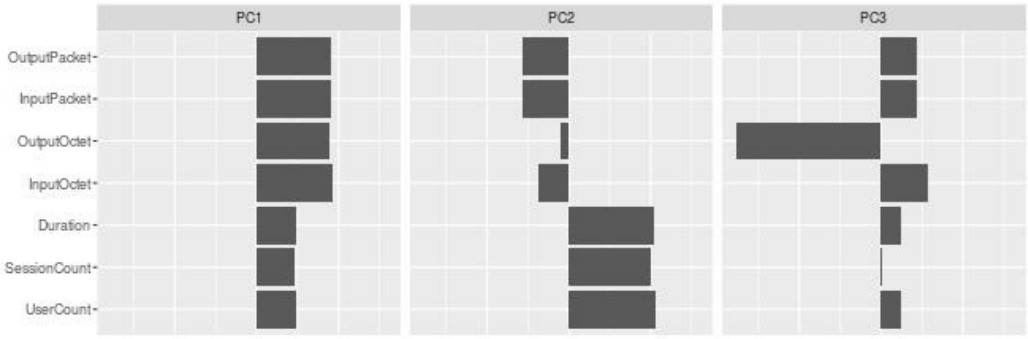


FIGURE 3.8: The behavior of the main features relative to the three principal components.

variables which best explain the phenomena and contain the greatest part of the entire information.

In this case the first three principal components bring the cumulative proportion of variance to over 95%. Figure 3.8 demonstrates the participation proportion of each feature to the principal components. We observe that the first principal component is associated with all the above features in a positive manner, more specifically with the usage attributes. The second principal component is decreasing with the usage attributes, and increasing with the density values. The single largest contributor to the third principal component is the output data in octet or the amount of downloaded bytes by the wireless users. The other features play less important roles in the third component, positively or negatively. Approximately categorizing the principal components like so, provides us with a deeper understanding of the connection of the aforementioned features, density or usage attributes, with the principal components resulted by PCA technique.

3.2.4 Summary

In this section we explored the large log data of RADIUS authentication collected from FEUP in approximately two years. We presented a summary of usage evolution through the academic semesters. Then we conducted a preliminary analysis for user sessions and for access points. Moreover, we performed some enumeration processes on the data to extract the main data features: User Count, Session Count, and Connection Duration as Density Attributes, and Input and Output Data in Octets and Packets as Usage Attributes. Eventually, we analyzed the correlation of the aforementioned features and performed PCA to select three principal components.

3.3 Testbed Deployment

In this section we explain how we deployed a Testbed with a single AP and generate a number of anomalies in a controlled environment for experimental purposes. We worked with FreeRADIUS server [98] which is widely used for Enterprise Wi-Fi and IEEE 802.1X network security and communication, particularly in the academic community, including Eduroam.

The Testbed is deployed in a home environment, with a single AP and 6 regular users and between 4 guest users. The experiment contains 6 weeks of data, 30 working days, and is performed in two different time span, once in November 2015 and

a while later in April 2016. There exist 20 normal days with no anomalies provoked, and 10 abnormal days containing at least one anomalous event a day. Each anomaly takes from 15 minutes to around an hour.

3.3.1 Server Configurations and Users Specifications

The FreeRADIUS server was setup on a Linux machine with 2.30 GHz Intel(R) Core(TM) i5-2410M CPU, and 8GiB System Memory. The database system which is used to store primary configurations and AAA information is MySQL and consist of 10 specified tables. The principal tables employed for data collection and analysis are labeled as *radcheck* (authentication), *radpostauth* (authorization) and *radacct* (accounting). Other essential configurations are conducted directly on FreeRADIUS setting files, such as server and client security configurations, required certificates, and database setups.

The AP is an enhanced 802.11g wireless access point powered by D-Link 108G technology, DWL-2100AP, and supports WPA and WPA2 security protocols. The wireless users connected to this network during one month of experiment consist of two laptops, two smart phones, and two tablets. A summary of the users' specifications in terms of devices, operating systems and participation time in the experiment is provided in Table 3.3. Obviously not all the users were present everyday and every hour of the test, but they follow a natural form of entering and exiting the network. Some devices were disassociated from the network when the users simply depart from the coverage area and others were deliberately disconnected in the time of specific anomaly generation. In the coming sections we present all types of anomalies generated and organized for this Testbed.

TABLE 3.3: A summary of the Testbed users' specifications

Device	OS	Participation Time
Surface Pro II	Win 10	All the time
Asus	Win XP	All the time
Alcatel onetouch	Andriod	Frequently
iPhone	iOS	Infrequently
iPad	iOS	All the time
Dell	Win 10	Infrequently

3.3.2 Network Anomaly Generation in a Controlled Environment

In this section we describe how some of the known wireless network issues are re-generated to make the desired data records for the evaluation of the proposed methodologies. In the course of the experiment we have two types of days: *Normal* and *Abnormal*. Users do not follow a script for their normal behaviour, and just use the network as they see fit. In abnormal days, however, one or some kind of anomalies are provoked to inspect the behavior of the model under abnormal circumstances. The anomalous patterns selected for this purpose are common cases that occur in real networks relatively often and affect the performance of users connection and availability of the network. Succeeding paragraphs deal with the specific aspects of these anomalies and point out how to replicate them.

AP Shutdown/Halt

An AP is considered to be shutdown for a while or halted when there is no session recorded in the accounting table. We generate this anomaly by turning off the AP power deliberately for some periods of time in different times of the day.

Heavy Usage

Single User We provoke this anomaly by making one user perform heavy downloads or uploads. It might affect the rest of the associated users depending on the amount of usage, duration, time of the day and other relevant factors.

Multiple Users This anomaly emerges when more than one user utilizing the network excessively, and therefore the overall throughput of the network intensifies. This could occur in a normal day or as an anomalous event, and the network tolerance varies for different networks and different AP configurations. In any case, the proposed model is expected to detect the irregularity and report the level of hazard so that the network managers can take control of the situation and make required changes if possible.

Wireless Network Interference

In a real network, a variety of things can interfere with the radio waves, degrading the quality of connection and decreasing the network reliability. Sources of interference are commonly from other wireless networks in the vicinity when they all locate in the same channel, from non-802.11 devices such as microwave ovens or cordless phones that use 2.4GHz band as well, from other clients in a crowded environment when they all try to transfer data at the same time, and from RF effects such as hidden terminals or capture effects.

In this work we intend to cause interference in a systematic and controlled manner. For this aim, we use a python script named wifijammer [99] to intentionally jam wireless clients or APs in the range to simulate the same outcome as the aforementioned interference. The jamming process works by sending one de-authentication packet to the client from the AP, one de-auth to the AP from the client, and one de-auth to the AP destined for the broadcast address to de-authenticate all clients connected to the AP. Many APs, however, ignore de-auth to broadcast addresses.

We employed wifijammer in the following plans by applying different configurations and creating different forms of interference.

Jamming the Entire Channel In this practice, the monitor mode interface is set to listen and de-authenticate clients or APs on a specific channel. This way of jamming influence all the available networks on the current channel and imply interference caused by busy channels.

Jamming Clients with Various Time Intervals Executing the de-authentication procedure with short time intervals hinder clients from recovering and disable them for the entire period of jamming, so the immediate result in the accounting table is a stop session from each client and then a silent period without any start session. However, de-authenticating with a larger time interval makes clients reclaim and try to get back the connection to the AP, and subsequently many short sessions are observed in the accounting table because they are de-authenticated right after getting

connected again. In such manner we can replicate two interference cases observed in the real data sets frequently: stop session followed by a silent period, and several consecutive short sessions.

Jamming Specific Clients De-authenticating some specific clients and not the rest resembles the hidden-terminal situation, when one client is forced to back-off and delay data transfer because the other clients can not sense its send-request. Depending on the time interval discussed earlier, the sessions outcome in the accounting table could be different.

3.3.3 Summary

In this section we described the deployment of a small Testbed with one AP and six wireless users in FreeRADIUS server. Further we disclosed the server configurations and users specifications in detail. Then we described the process of anomaly generation in this real Testbed. The anomalous cases reproduced contain AP Shutdown/Halt, Heavy Usage, and Interference.

3.4 Wireless Network Simulation

When acquiring ground truth is too expensive and time-consuming, network simulation seems to be an effective solution to achieve a close to reality setup that is computationally tractable. After exploring various simulation frameworks such as NS3 [90] and OPNET [91], we chose to use OMNeT++ [92] and INET Framework [93] for our research purposes. Besides the well-structured framework and user-friendly IDE that facilitate analysis and data gathering, OMNeT++/INET provides an adequate set of modules supporting physical and radio models for 802.11.

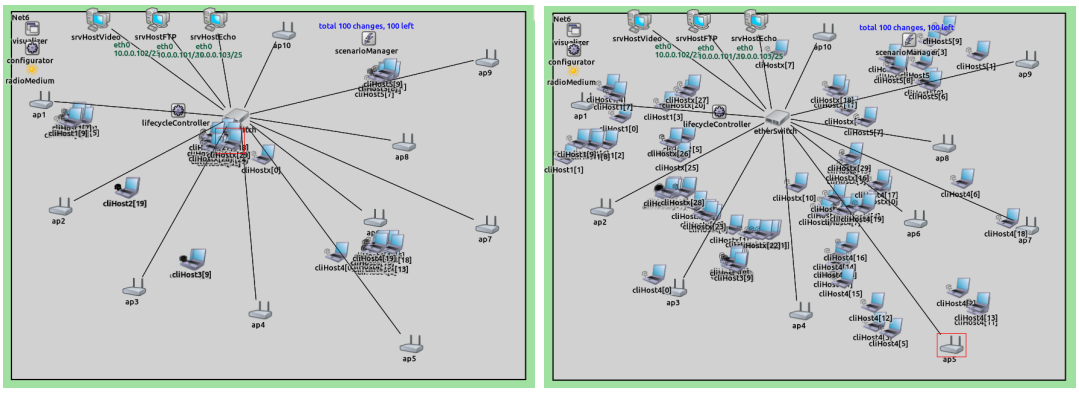
3.4.1 Simulation Setup

We performed an extensive set of simulations using OMNeT++ [92] simulator and INET framework [93]. OMNeT++ is a C++-based discrete event simulator (DES) for modeling communication networks, multiprocessors and other distributed or parallel systems. It has a generic architecture and is used in various problem domains including the modeling of wired and wireless communication networks.

One of the major network simulation model frameworks for OMNeT++ is the INET Framework that provides detailed protocol models for TCP, IPv4, IPv6, Ethernet, Ieee802.11b/g, MPLS, OSPFv4, and several other protocols. We used OMNeT++ along with INET Framework to simulate the IEEE 802.11 WLANg (2.4 GHz band) in infrastructure mode.

As in any discrete event simulator, in OMNeT++, events take place at discrete instances in time taking zero time to happen. It is assumed that nothing important happens between two consecutive events. Thus the simulation time is relevant to the order of events in the events' queue, and it could take more than the real CPU time or less than that based on the number of nodes, amount of traffic, and other details of the network.

With the current number of nodes (10 APs and 100 STAs) and our chosen traffic plan, 10 minutes of simulation time took around 2 hours of CPU time. Our HMM approach operates on 40 consecutive time-slots of 15s simulation time each.



(A) The initial picture of the network (B) After 30s of simulation

FIGURE 3.9: The initial picture of the wireless network simulated in OMNeT++/INET, and the location of the wireless stations after 30s of simulation.

The simulation consists of one normal scenario and four anomalous scenarios: AP Shutdown/Halt, AP Overload, Noise, and Flash Crowd. We simulated 15 instances of 3000s simulation time for normal scenario, and 5 instances of 3000s simulation time for each of the anomalous scenarios to have enough data for training and test sets.

Normal Scenario

Figure 3.9a shows the initial picture of a normal scenario, the location of the access points (APs), wireless stations (STAs), and the servers. Figure 3.9b displays how the stations scatter in the wireless ground after passing 30 s (simulation time) from the beginning of the simulation.

In the normal scenario, there are 10 APs and 100 STAs. Each STA is initially associated to one of the available APs depending on its location. During the simulation, wireless stations - based on their mobility models - are handed over to other APs when moving around the simulation ground. Then, according to the defined traffic plans in section 3.4.2, each node sends and receives packets to the existing servers.

3.4.2 Mobility Models of the Wireless Stations

The APs are stationary and the wireless nodes follow different mobility patterns. In the current experiment, the mobility models of the nodes are selected in a way to emulate the usage behavior of three typical places in a campus.

The mobile nodes initially connected to one AP follow the *Linear Mobility* pattern which is configured with speed, angle and acceleration parameters (Table 3.4). The mobile nodes move to random destinations with the specified parameters and when they hit a wall they reflect off the wall at the same defined angle. These nodes connect to another AP in the vicinity, and sometimes they lose the connection when they move to blind spots. This pattern is selected to symbolize the nodes with some degree of freedom but within a limited space like administrative offices. In this experiment 20% of the wireless stations follow this mobility model.

The nodes following the *Mass Mobility* model move within the room. This pattern of mobility is intended to represent places like classroom or library in which users

TABLE 3.4: Wireless nodes' specifications in terms of mobility models

Mobility Model	# Nodes	Mobility Parameters
Linear Mobility	20	speed: truncnormal[1] (20mps, 10mps) angle: normal[2] (270deg, 90deg) acceleration: 0
Mass Mobility	30	speed: truncnormal(70mps, 50mps) changeInterval: truncnormal(2ms, 0.5ms) changeAngleBy: normal(90deg, 90deg)
Random Waypoint Mobility	50	speed: uniform[3] (50mps,50mps) waitTime: uniform(3s,8s)

do not frequently leave the place, but still have some motions in the place. 30% of the nodes follow this mobility pattern.

The rest of the wireless nodes follow the *Random Waypoint Mobility* and move to a random destination (distributed uniformly over the playground) with a random speed. When nodes reach their target position, they wait for a specified *waitTime* and select a new random position afterwards. This type of movement resembles the random mobile users around the wireless ground mostly connected with their mobile devices. Half of the nodes in this experiment follow this mobility pattern.

A summary of wireless nodes' specifications in terms of mobility models is provided in Table 3.4.

Traffic Generation

As it is shown in Figure 3.9, there are three main servers wire-connected to the Ethernet switch: srvHostVideo, srvHostFTP, and srvHostEcho. The traffic transferred between wireless stations and the servers (through APs) is considered to be User Datagram Protocol (UDP). The video server (srvHostVideo) sends UDP packets with the message length of $\mathcal{N}(600B, 150B)$ to a number of clients resembling the video downloading by those users. The FTP server (srvHostFTP) is to receive the FTP uploads of some clients with message length of $\mathcal{N}(500B, 100B)$. The other server (srvHostEcho) is in charge of both sending and receiving traffic to all the users. This traffic pattern represents web browsing and email checking by all the wireless users. The echo packets length are configured to be smaller than the previous ones, $\mathcal{N}(200B, 50B)$, indicating lighter traffic transmission. 35% of the users download via srvHostVideo, and 20% of them upload via srvHostFTP. In *AP Overload* anomalous scenario one more server is added to take care of heavy channel utilization (srvHostBurst), and more details about it can be found in section 3.4.3.

Path Loss Models

As the signal propagates through space its power density decreases. Path loss might be due to the combination of many effects, such as free-space loss, refraction, diffraction, reflection, and absorption. The path loss model computes the power loss factor based on the traveled distance, the signal frequency and the propagation speed. In our experiments we utilized the following four path loss models to increase the complexity of the simulation and make it more realistic:

- Free Space Path Loss: is the loss in signal strength resulting from a line-of-sight path through free space, with no obstacles nearby to cause reflection or diffraction.
- Log Normal Shadowing: is a stochastic path loss model, where power levels follow a lognormal distribution. It is useful for modeling shadowing caused by objects such as trees.
- Rician Fading: is a stochastic path loss model which assumes a dominant line-of-sight signal and multiple reflected signals between the transmitter and the receiver. It is useful for modeling radio propagation in an urban environment.
- Rayleigh Fading: is the loss in signal magnitude according to a Rayleigh distribution - the radial component of the sum of two uncorrelated Gaussian random variables. It is useful for modeling the effect of heavily built-up urban environments on radio signals.

3.4.3 Anomalous Scenarios

AP Shutdown/Halt

When there is no session recorded for a given AP in RADIUS accounting table in a period of time, it is likely that the AP has stopped working - possibly due to a technical problem or power failure. In our simulation, we reproduced this anomaly by turning off the AP power deliberately during the *halt-period* for some *time-slots*. We used *ScenarioManager* in OMNeT++/INET and *shutdown* and *startup* commands to turn off the AP module and start it over after some periods of time.

AP Overload

In this anomalous case, excessive channel utilization occurs that could be the consequence of excessive download or upload by a number of wireless users. In such circumstances, the clients get disconnected from the current AP frequently even with the presence of high signal strength. In this experiment we simulated AP heavy usage caused by all of the users of a given AP. Burst server (srvHostBurst) sends UDP packets to the given IP addresses in bursts during the *burst-duration* period which resembles the heavy downloads of the wireless users. In the *sleep-duration* period the burst flow stops and the channel utilization gets back to normal. This experiment contains three different cases as following:

- burst-duration $<$ sleep-duration.
- burst-duration $=$ sleep-duration.
- burst-duration $>$ sleep-duration.

Noise

Thermal noise, cosmic background noise, and other random fluctuations of the electromagnetic field affect the quality of the communication channel. This kind of noise doesn't come from a particular source, nor propagates through space. If the noise level is too high, the signal strength will degrade and the performance will decrease.

In the current experiment we change the level of noise power by adjusting the value of *IsotropicBackgroundNoise* parameter in the simulator. The default value of

this parameter is set to -110dBm which is the minimum noise level in Wi-Fi networks 802.11 variants. We gradually increase the noise power to -90dBm and record the simulation results repeated 10 times for each experiment. According to the study in [100], the average noise level in a busy university campus had a stable value at around -94 dBm.

Flash Crowd

In wireless networks an unexpected surge of traffic occurs mostly due to the beginning or ending of an event when the majority of the wireless users abruptly enter or leave a place and consequently associate to or disassociate from an AP. Such incidents are not necessarily an anomaly in terms of performance or connectivity issues, but could be considered more as a sudden change to a routine network. To see whether the proposed models are able to detect such alterations in the normal usage pattern, we simulate this example in two experiments:

- Arrival: simultaneous association of 10 new nodes to the current AP.
- Departure: simultaneous disassociation of 10 existing nodes from the current AP.

3.4.4 Summary

In this section we represented the simulation of 10 AP and 100 wireless stations in infrastructure mode in OMNeT++/INET. We defined the characteristics of our simulation such as simulation time, mobility models of the wireless nodes, traffic generation, and path loss models. Finally we described the generated anomalous scenarios and their specifications. Simulated anomalous scenarios consist of AP Shutdown/Halt, AP Overload, Noise, and Flash Crowd.

3.5 Conclusion

In this chapter we described the different data sets that will be used in the models described in the following chapters. Although having different sources, these data sets are all related to the wireless station RADIUS authentication data collected at access points.

We explored the large log data of RADIUS authentication collected from FEUP in approximately two years, and presented a summary of usage evolution through the academic semesters. Then we conducted a preliminary analysis for user sessions and access points. Moreover, we performed enumeration processes on the data to extract the main data features: User Count, Session Count, and Connection Duration as Density Attributes, and Input and Output Data in Octets and Packets as Usage Attributes. Yet we analyzed the correlation of the aforementioned features and performed PCA on them to select three best features.

Further, we described deployment process of a small Testbed with one AP and six wireless users in FreeRADIUS server, and disclosed the server configurations and users specifications in detail. Then we described the process of anomaly generation in this real Testbed. The anomalous cases reproduced contain AP Shutdown/Halt, Heavy Usage, and Interference.

Finally, we represented the simulation of 10 AP and 100 wireless stations in infrastructure mode in OMNeT++/INET. We defined the characteristics of our simulation such as simulation time, mobility models of the wireless nodes, traffic generation, and path loss models. Ultimately we described the generated anomalous scenarios and their specifications. Simulated anomalous scenarios consist of AP Shutdown/Halt, AP Overload, Noise, and Flash Crowd.

Chapter 4

Selected Approaches to Hidden Markov Modeling for Anomaly Detection and Pattern Recognition

4.1 Introduction

We intend to inspect and characterize the usage pattern of wireless networks and their inherent dynamics in order to provide models for anomaly detection. For this purpose we explore the temporal usage behavior of the network by applying various types of Hidden Markov Models. We observe the usage pattern of up to 100 APs in one week period in 2011 at the Faculty of Engineering of the University of Porto. The first step of this study consists of constructing Hidden Markov Models from 802.11 AP usage data. We then apply statistical techniques for outlier detection and justify the presented outliers by inspecting the models' parameters and a set of HMM indicators. The reason for this justification is that there is no explicit ground truth provided for the RADIUS data set (3.2) that would conveniently lead us to anomalies. We finally introduce examples of wireless networks anomalous patterns based on the transitions between HMM states and provide an analysis of the entire set of APs under study.

In this chapter we proceed as follows. We first provide some background information on connectivity problems in 802.11 wireless networks and Hidden Markov Models application for anomaly detection purposes in these networks in section 4.2. In section 4.3 we present Hidden Markov Models definition, parameters, and key problems in subsection 4.3.1, selected approaches to HMM modeling in subsection 4.3.2, and outlier detection methods by the state of the art and HMM techniques in subsection 4.3.3. In section 4.4 we provide some discussions on outliers analyzing the outlier quality indicators in subsection 4.4.1, and investigating anomalous patterns as collective outliers in subsection 4.4.2. In section 4.5 we present the experimental results on HMM outliers in subsection 4.5.1, and anomalous patterns in subsection 4.5.2. In both above mentioned subsections, the experimental results are demonstrated through a case study and a systematic analysis of the entire set of data. In section 4.6 we present concluding remarks.

4.2 Background

The question of performance in 802.11 wireless networks becomes increasingly important as many new emerging applications such as mobile information access, real-time multimedia communications, and cooperative work require sufficient bandwidth and consistent connectivity. Some major circumstances leading to performance degradation in such networks are contention and collision, rate diversity and fairness, random losses, and TCP performance and traffic asymmetry [5, 101]. Due to such fundamental issues of the wireless medium, users of 802.11 networks experience a number of connectivity problems such as authentication failures, intermittent connections to 802.11 APs and inconsistent or lack of coverage. These connectivity problems can also be the consequence of RF interference, weak RF or RF holes, and users associating to overloaded APs [102].

To detect and address such anomalies in 802.11 networks we apply Hidden Markov Models which are frequently used in network measurements to obtain temporal information of signals in the network. Hidden states of HMMs encode different probability distributions. For example, states for *low* and *high* network activities and their respective observation distributions. A statistical learning methodology estimates automatically the states' parameters and the state transition probabilities from the previous observations. The typical approach for using HMMs in network-related work is to automatically learn an HMM for each behavior or class of network activities [103, 104]. In this work we detect anomalous time instances by analyzing the likelihood series of three approaches to HMM modeling: a single model for all the network data, separate models for each AP's data, and groups of HMMs by mixture estimation technique. We justify and evaluate the anomalies detected by each model through HMM parameters exploration and analysis. Furthermore, we propose a number of network anomalous patterns deduced from states transition sequences and present the potential HMM variations capable of detecting specific types of patterns.

4.3 Methodologies

In circumstances where acquiring labeled data is troublesome or time-consuming, unsupervised techniques are most widely applicable to discover underlying patterns of the data. In the current case, there is no explicit ground truth provided for the RADIUS data set (3.2) which would conveniently lead us to anomalies. Therefore, we made use of well-known outlier detection techniques as well as various HMM approaches to investigate this demanding network management work.

4.3.1 Hidden Markov Model

An HMM is a doubly stochastic process with an underlying stochastic process that can only be observed through another set of stochastic processes that produce the sequence of observed symbols [105].

An HMM is completely defined by the following parameters:

- The number of hidden states, N.
- The discrete set of hidden states, $S = \{s_i\}, 1 \leq i \leq N$.

- If observations are discrete, the number of possible observations, M, and the discrete set of such observations, $V = \{v_k\}, 1 \leq k \leq M$.
- If observations are continuous, the corresponding dimensionality of the observation space, d.
- The state transition probability distribution, $P(h_t|h_{t-1})$, represented by a matrix $A = [a_{i,j}], 1 \leq i, j \leq N$, where $a_{i,j} = P(h_t = s_j|h_{t-1} = s_i)$.
- The emission probability distribution, $P(o_t|h_t)$. For discrete observations, this is represented by a matrix $B = [b_i(v_k)]$, $1 \leq i \leq N$, $1 \leq k \leq M$, where $b_i(v_k) = P(o_t = v_k|h_t = s_i)$. For continuous Gaussian observations, the emission probability density is defined by the set of d-dimensional means, $\mu = \{\mu_i\}$, and the set of $d \times d$ covariance matrices, $\Sigma = \{\Sigma_i\}, 1 \leq i \leq N$.
- The initial state probability distribution, $P(h_0)$, represented by a vector $\pi = [\pi_i], 1 \leq i \leq N$, where $\pi_i = P(h_0 = s_i)$.

An HMM λ models the joint distribution $P(O, H|\lambda)$ of a sequence of hidden states $H = (h_0, h_1, h_2, ..., h_T)$ and a sequence of observations $O = (o_1, o_2, ..., o_T)$ as:

$$P(O, H|\lambda) = P(h_0) \prod_{t=1}^{T} P(h_t|h_{t-1}) P(o_t|h_t), \tag{4.1}$$

where, on the right-hand side, we omit the dependency on λ to simplify notation. Furthermore, it is often assumed that both distributions $P(o_t|h_t)$ and $P(h_t|h_{t-1})$ are stationary, that is:

$$P(o_t|h_t) = P(o_{t'}|h_{t'}), \tag{4.2}$$

$$P(h_t|h_{t-1}) = P(h_{t'}|h_{t'-1}), \text{ for all } t' \tag{4.3}$$

Using the model λ, an observation sequence $O = o_1, o_2, ..., o_T$ is generated as follows:

1. Select an initial state, h_1, according to the initial state probability distribution, π;
2. Set $t = 1$;
3. Choose o_t according to observation probability distribution in state h_t, $b_{h_t}(k)$;
4. Choose h_{t+1} according to the state transition probability distribution for state h_t, $a_{h_t,h_{t+1}}$
5. Set $t = t + 1$; return to step 3 and continue until $t = T$

Given the form of the HMM, there are three key problems whose solution is interesting for real world applications. These problems are listed as following [105]:

Problem 1 – Given the observation sequence $O = o_1, o_2, ..., o_T$ and the model $\lambda = (A, B, \pi)$, compute $P(O|\lambda)$, the probability of the observation sequence. (Forward-backward)

Problem 2 – Given the observation sequence $O = o_1, o_2, ..., o_T$, choose a state sequence $S = s_1, s_2, ..., s_T$ which is optimal in some meaningful sense. (Viterbi)

Problem 3 – Given the observation sequence $O = o_1, o_2, ..., o_T$ and the model $\lambda = (A, B, \pi)$, adjust the model parameters $\lambda = (A, B, \pi)$ to maximize $P(O|\lambda)$. (Baum-Welch)

The compact notation $\lambda = (A, B, \pi)$ defines an HMM with discrete emission, and $\lambda = (A, \mu, \Sigma, \pi)$ represents an HMM with continuous Gaussian emission.

The forward-backward algorithm provides an efficient solution to the evaluation problem (likelihood of an observation sequence O given the λ model), and also scoring problem (choosing the best λ model among the competing models). The forward variable $\alpha_t(i) = P(o_1 o_2 ... o_t, h_t = s_i|\lambda)$, and the backward variable $\beta_t(i) = P(o_{t+1} o_{t+2} ... o_T|h_t = s_i, \lambda)$ are computed recursively according to Equations 4.4 - 4.7 as follows:

$$\alpha_1(i) = \pi_i b_i(o_1) \qquad 1 \leq i \leq n \tag{4.4}$$

$$\alpha_{t+1}(j) = \left[\Sigma \alpha_t(i) a_{ij} b_j(o_{t+1})\right] \qquad 1 \leq j \leq n, \quad 1 \leq t \leq T-1 \tag{4.5}$$

$$\beta_T(i) = 1 \qquad 1 \leq i \leq n \tag{4.6}$$

$$\beta_1(i) = \Sigma a_{ij} b_j(O_{t+1}) \beta_{t+1}(j) \qquad 1 \leq i \leq n, \quad 1 \leq t \leq T-1 \tag{4.7}$$

These probabilities contain very small values, due to the multiplication of many transition and emission probabilities all below 1. As the length of the sequences increase, the forward-backward probabilities will likely exceed the available machine precision. Consequently to avoid the underflow issue the scaled values must be substituted. The exact formulations of the scaled version of forward and backward variables can be found in [83, 106].

According to forward-backward algorithm there are T ways to compute the likelihood value or probability of the observation sequence O given the model λ.

$$P(O|\lambda) = \sum_{j=1}^{n} \alpha_t(j) \beta_t(j) \qquad 1 \leq t \leq T \tag{4.8}$$

for $t = T$ the above equation will be in the following form:

$$P(O|\lambda) = \sum_{j=1}^{n} \alpha_T(j) \beta_T(j) \tag{4.9}$$

Substituting the forward recursion of 4.8 for $\alpha_T(j)$ gets the following equation which will be further used in analysis presented in chapter 6.

$$P(O|\lambda) = \sum_{j=1}^{n} \sum_{i=1}^{n} \alpha_{T-1}(i) a_{ij} b_j(o_T) \beta_T(j) \tag{4.10}$$

Once again due to the vanishingly small likelihood probabilities produced in long time-series, the logarithmic value is used as log-likelihood values.

4.3.2 Selected approaches to HMM Modeling

The HMMs we use in this chapter are multivariate, having 3 main features, consist of continuous Gaussian distribution, and contain fully connected states, thus transitions are allowed from any state to any state.

HMMs consist of states that encode different probability distributions. For the current work, we considered 3 states of *low*, *medium* and *high* addressing the usage (load) of the APs and the respective observation distributions.

In this section three different HMM modeling approaches are provided to characterize the usage pattern of the entire set of APs. All types of HMMs in this work retain 3 states of *low*, *medium* and *high*.

Separate Models per AP

To induce HMMs specifically for each AP, a vector quantization process is performed on the observation data and 3 clusters (low, medium and high) are produced by k-means. Hence, the emission matrix of each model is formed by estimating the distribution parameters of the clusters. Moreover, the transition matrix is a result of an enumeration procedure on the sequences of observed states which are labeled base on the assumed distribution parameters.

Single Model for all APs

In this approach the observation sequences of all APs participate in generating a single HMM which characterizes the entire set of data in one model. The process of the formation and estimation of states' distribution parameters and transition probabilities are similar to the previous model with this difference that the states distributions and the transition probabilities are estimated using the expanded set of data. The single model is expected to be more robust using the data belonging to all APs, while the separate models might be better cater for the specificities of each AP.

Groups of APs and Mixture of HMMs

Given a source of time-series data, it is often advantageous to determine whether there are qualitatively different regimes in the data and characterizing those regimes. HMMs have been shown empirically to be capable of modeling the structure of the generative processes underlying numerous types of real world time-series. The mixture modeling is based on the well-established method of Expectation Maximization (EM) for estimating mixture parameters from the set of data. A mixture model [107] is defined as following:

$$P(O_i|\Theta) = \Sigma_{k=1}^{K} \alpha_k P(O_i|\lambda_k) \tag{4.11}$$

The mixture probability density function (pdf) is parameterized by $(\alpha_1, ..., \alpha_k, \lambda_1, ..., \lambda_k)$, consist of the prior probabilities $\alpha_k, k = 1, ..., K$, and the likelihood function of the HMMs denoted by $P(O_i|\lambda_k)$. The λ_k is the set of parameters that describe the density functions of linear HMMs with multivariate emission distributions. The observed data O_i then corresponds to the multi-dimensional time-series that reflect the underlying usage pattern of the APs. The goal is to maximize Equation 4.11 by choosing optimal parameter set. This problem is generally solved by the EM algorithm which finds a local optimum for the above function. The outcome is the groups of APs with one optimized HMM as the representative of the group.

The mixture method concisely performs the following main steps, given a collection of K initial HMMs $\lambda_1^0, ..., \lambda_K^0$:
1) Iteration:

- Generate the initial groups of sequences by assigning each sequence O_i to the model k for which the likelihood is maximal.

- Calculate new parameters for each model $\lambda_1^t, ..., \lambda_K^t$ using re-estimation algorithm (*Baum-Welch*) based on their current parameters $\lambda_1^{t-1}, ..., \lambda_K^{t-1}$ and the assigned weights of the participating sequences.

2) Stop: If the improvement of the objective function is below a given threshold ϵ, the grouping of the sequences does not change or a given iteration number is reached.

Our assumption is that grouping APs and presenting an optimized HMM per group, has the potential to enhance the quality of the group models to conquer the weak points of the very generalized or very specific models (single HMM vs. separate HMMs). We show results that validate our assumption in Section 4.5

4.3.3 Outlier Detection Methods

Outlier detection, also referred to as anomaly detection, event detection, or deviant discovery, is the process of distinguishing observations that lie outside the regular pattern of a distribution and do not comply with the well-defined expected behavior [108].

Univariate Outliers: Feature by Feature

Individual data instances which are not compatible with the normal pattern of the rest of the data are called point anomalies. Point anomalies are the simplest form of anomalies detected as they lie outside the boundary of the normal zones. They can be single points each with a different pattern or small regions composed of several point anomalies.

To detect univariate outliers, features are inspected one by one without considering any correlation between them. Thus, any instance out of the normal boundary could be marked as an outlier without looking at the other accompanying features. In this work, univariate outlier detection is performed using *boxplot.stats* function from R, which returns the statistics for producing *boxplots* [109]. It labels the data points lying beyond the extremes of the *box-and-whisker* plot. An argument of coefficients is used to control how far the whiskers extend out from the box of a boxplot. We assumed $coef = 3$ to get the most extreme values as outliers. In an example AP (AP#0), there is no outlier detected in the first two features, and only 1 extreme is observed in the third feature.

Multivariate Outliers: 3D Impression of Data

Assuming the data is multivariate normally distributed in D dimensions, the *Mahalanobis* [110] distance of such set of data follows a *Chi-Square* distribution with D degrees of freedom. There are two approaches for outlier detection using Mahalonobis distances. The first one marks observations as outliers if they exceed a certain quantile of the chi-squared distribution. The second is an adaptive procedure looking for outliers specifically in the tails of the distribution, beginning at a certain *chisq-quantile*.

For this purpose the *aq.plot* function [111] from *mvoutlier* package [112] in R is used which plots the ordered squared robust Mahalanobis distances of the observations against the empirical distribution function of the MD_i^2 (squared Mahalanobis distance). Figure 4.1 shows the outliers detected based on the assumed quantile (97.5%) and the adapted quantile for the 3D data of AP#0. The points marked as X (in red) are the outliers projected on their two most robust principal components.

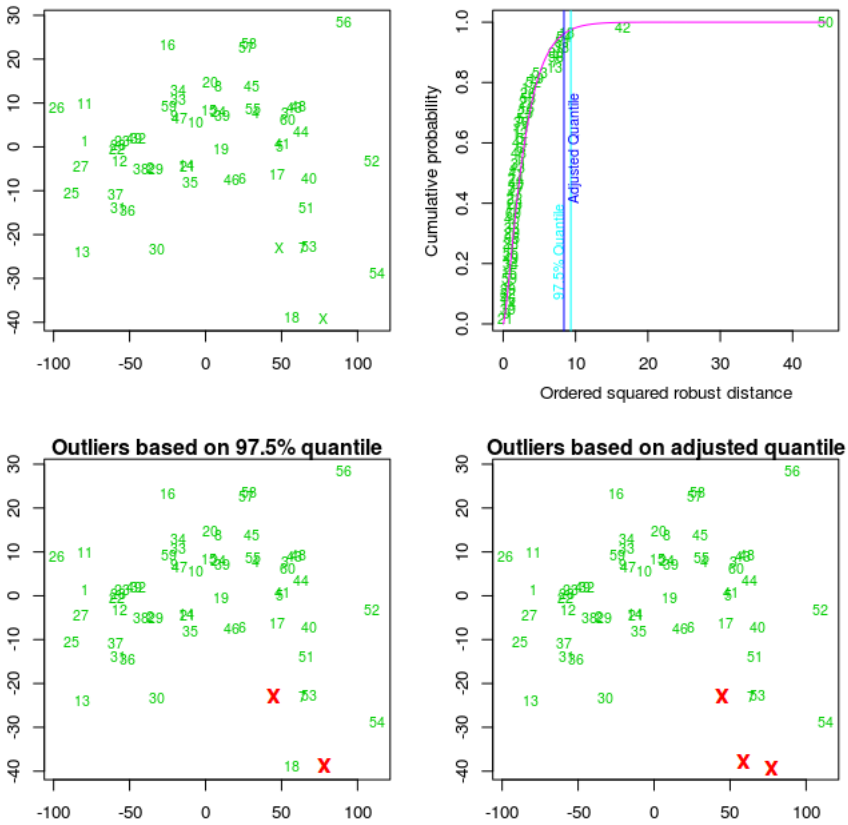


FIGURE 4.1: Multivariate Outliers Detected for 3D Data of AP#0

The first and second approaches detect 2 and 3 outliers shown in Figure 4.1 bottom-left and bottom-right, respectively. The multivariate outliers of the experimental results in Section 4.5 are identified using the first approach.

Temporal Outliers: Time Series

Contextual or conditional anomalies occur when a data instance is anomalous in one specific context and not in the others. The notion of context (or vicinity) determines the structure of data which has to be considered joint with data attributes to distinguish the anomalies. A salient example of a contextual attribute forming the data context is time in time-series data. Contextual anomalies have been mostly explored in temporal data [113, 114] and spatial data [115].

Accordingly, the third type of outliers inspected for the current data set is the temporal outliers. Thereupon, *stl* function [116] from R is employed which decomposes a time-series into seasonal, trend and irregular components using *loess* smoother [117]. As the data set of this work is inherently periodic, containing 12 hours a day for 5 consecutive working days, a time-series object with frequency of 12 is built and given to *stl* function. Figure 4.2 shows the time series data of AP#0, the seasonal, trend and remainder component. The X point is the only temporal outlier detected for the data of AP#0.

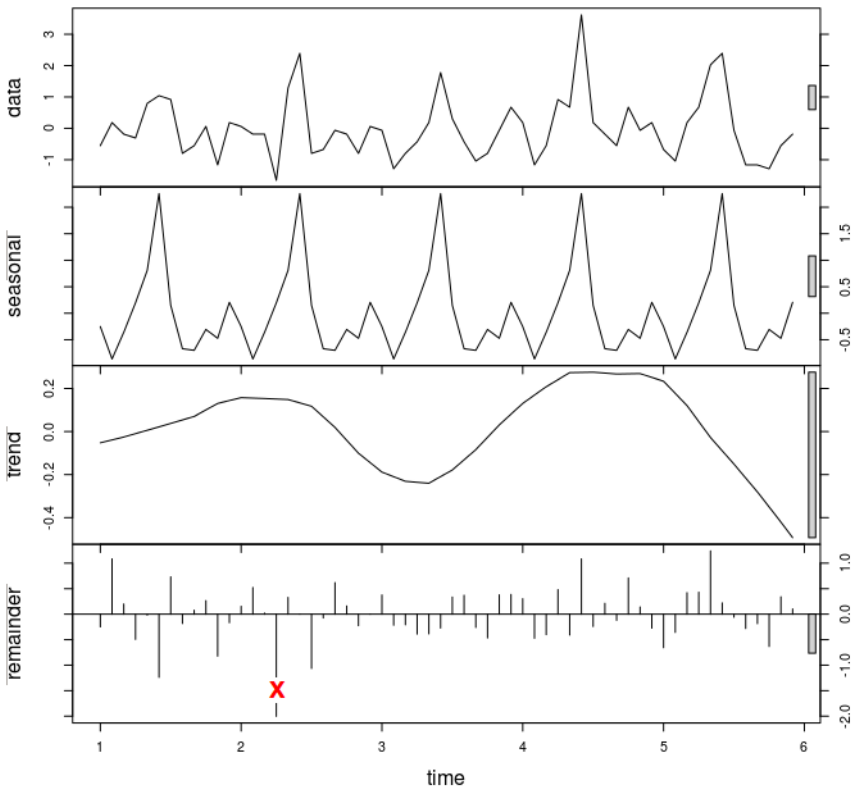
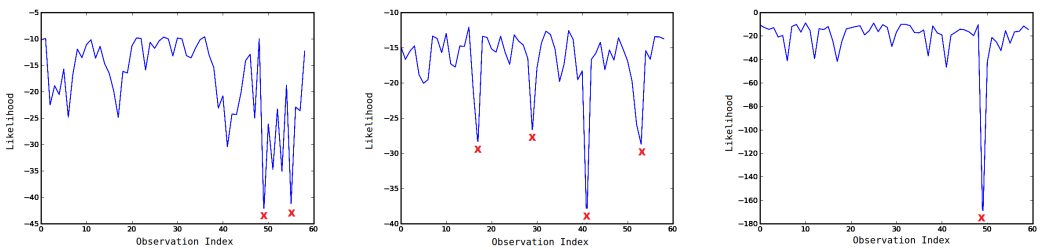


FIGURE 4.2: Temporal Outliers Detected for Data of AP#0



(A) Separate HMMs (B) Single HMM (C) Mixture of HMMs

FIGURE 4.3: Likelihood Series of Three Variations of HMMs

Hidden Markov Models: Likelihood Series

Measuring the likelihood of each observation instance (o_t, $1 \leq t \leq T$) in an observation sequence $O = o_1, o_2, ..., o_T$ given the HMM model, produces a likelihood series of the entire observation sequence. In the likelihood series some values are basically out of the normal range of the rest of the series, hence simple outlier detectors indicate them as outliers. Figure 4.3 demonstrates the likelihood series of AP#0 for the separate, single and mixture HMM models. The anomalies are marked with red X. As this figure shows some points are detected by more than one HMM model, for instance separate and mixture models both detect point #49 as outliers.

4.4 Discussion on Outliers

4.4.1 Outlier Quality Indicators

In this section we investigate some of the salient properties of HMMs that explain the likelihood results of different HMM models and provide justifications to substantiate the detected outliers. In the absence of ground truth these indicators could provide the network manager with more robust assessment of detected anomalies.

Large Distance from the Assigned Hidden State

Given an HMM λ and an observation sequence of $O = o_1, o_2, ..., o_T$, the most probable set of states are generated by *Viterbi* algorithm as $O = s_1, s_2, ..., s_T$, $s_i \in S$. To estimate the distance of a data record in time t to its closest HMM state (s_t) in Viterbi path, we employed the concept of distance of a data point to a distribution. For this purpose *Mahalonobis* distance is calculated for each data point to its equivalent assigned state in the Viterbi path. To highlight the isolated records in terms of separation from the appointed HMM states, the univariate outlier detection method is utilized.

Less Likely State Transition

Intuitively we expect that the highest transition probabilities are observed between identical states (s_i to s_i), and the lowest probabilities between the most distant states (s_0 and s_2 in a 3 states HMM). The medium state (s_1) is the most uniformly distributed state in terms of transition probabilities to the higher (s_2) and lower (s_0) states. Having observed the regular HMM state transitions of the HMM variations, the least frequent transitions are more likely to be among the anomalous instances. For example when rare state modifications appear in Viterbi path or there exist no transition when according to the transition matrix it is expected to be, the chances of encountering an anomaly is higher. To distinguish the least likely state transitions, the HMM transition matrix is analyzed and transitions probabilities below 10% are marked as outliers.

Unbalanced Separation of the HMM States

Continuous observations indicate different distribution parameters for each hidden state. To determine the distances between HMM states, we utilized the *bhattacharyya.dist* function [118] from R calculating the distance between distribution pairs. Successive fluctuation between states could be a symptom of network malfunctioning. However, if the distance between two alternating states is not large enough the possibility of a network connectivity problem is less.

Other advantage of studying distance between HMM states is to understand the separation of states and declare the anomalous patterns with more confidence. For example when in a model (s_1) and (s_2) are very close, the anomalous pattern of *2020* is very similar to *1010*, or the anomalous pattern of *2121* does not necessarily reflect any changes. Therefore, we should inspect specific patterns based on distance between their states, and also take into consideration similar patterns.

We must carefully consider that HMM indicators and HMM outlier detectors deal with two different aspects of the HMM models. HMM outliers come from the abnormal likelihood values in the likelihood series, which reflect the trait of each

and every observation instance as well as the overall quality of the model. On the other hand, HMM indicators monitor the characteristic of each observation instance concerning the model parameters which determine whether there is any sign of abnormality associated with observation instances and assert the potential source of the problem from the model parameters' viewpoint. For example a data point is marked by *less likely state transition* HMM indicator because of its low transition probability which is one of the model parameters. Hence, according to the proposed model this point is suspicious. If the model has already reported the same point as outlier (based on its likelihood value), then we claim a true match. The likelihood outliers that are not confirmed by any of the model indicators are supposed to be false positives of the model.

4.4.2 Anomalous Patterns: Collective Outliers

Our intuition is that the normal usage pattern observed in APs consists of one or two peaks a day. Regardless of the AP location category (classroom, auditorium, administrative office, cafeteria, etc.) a gradual transition between successive hidden states is expected during the normal periods. There exist a number of anomalous patterns where a sequence of actions occur together. In HMM terminology, the anomalous patterns could be defined as the occurrences of a number of state transitions in a specific order. For example a high state transition to a low state and remaining there for some successive time slices, could indicate a special type of anomaly interpreted by network administrator. From the AP point of view, we have defined anomalous patterns as falling into one of the following categories [51]:

AP Halt/Crash

When the HMM is in the low state after a drop from a higher state and remains low for some successive time-slots (*210000* or *110000* or *220000*). Such patterns indicate possibility of AP halt or crash.

Persistent Interferences

Where repeated downturns are observed for a given AP in a day, possibly due to RF interference and RF holes. Decline from a higher state to the lowest state (s_0) more than three times a day indicates such anomalies (*10..10..10* or *20..20..20*).

AP Overload

In the presence of high HMM state right after a low state and intermittent fluctuation between the two states (*0202* or *0101* in case states 0 and 1 are distant or *1212* in case state 1 and 2 are distant). This could be the case of heavy utilization by some few users.

4.5 Experimental Results

In this section we discuss two principal lines of the current study: 1) HMM outliers, and 2) anomalous patterns. The experimental results are presented through a case study and a systematic analysis of the entire set of data. The data used for the following experiments is a subset of the dataset described in Section 3.2 for a period of

TABLE 4.1: Detected Outliers of AP#0 by All the Outlier Detection Techniques (HMM Indicators: Large distance from the assigned states (Dist Ind.), Less likely state transition (Prob Ind.), Rare transition probability (Trans Ind.)

	Day 2												Day 4												Day 5											
Day Hours	8	9	10	11	12	13	14	15	16	17	18	19	8	9	10	11	12	13	14	15	16	17	18	19	8	9	10	11	12	13	14	15	16	17	18	19
STOA																																				
Univariate																			X																	
Multivariate																			X							X										
Temporal					X																															
Separate HMMs																																				
Dist Ind.	*																									*						*				
Prob Ind.																																				
Trans Ind.																																				
Single HMM																			X																	
Dist Ind.	*																																			
Prob Ind.																																				
Trans Ind.																																				
Mixture of HMMs																			X							X										
Dist Ind.							*												*											*						
Prob Ind.																										*										
Trans Ind.																										*	*									

one week between 2011-12-12 and 2011-12-19. The summary of the relevant tests are provided in Table 4.1-4.4.

4.5.1 HMM Outliers

Case Study

In this part one AP is selected (AP#0) and its outliers are detected by various detection techniques (State Of The Art (STOA) and HMM models). Table 4.1 provides a comparative outlook of all detected outliers of this example by the different detection techniques. Detected anomalies are marked by X and those supported by HMM indicators are marked by *. The learning period consist of 5 working days and 12 hours per day, from 8:00 to 19:00. No outlier was observed in day 1 and day 3.

There are some hours which are marked by a majority of the detectors and those that are pointed out only by one or two detectors. For example one data point is labeled as outlier on day 4 at 13:00 by 4 out of the 6 detectors. Inspecting the data, the third feature (user count) is out of the normal range, so the entire record is marked by univariate detector. It is also detected by multivariate technique which consider the Chi-Square of Mahalonobis distances, but not by the temporal detector. The mixture HMM detects this point as well as the single HMM, while the separate model do not.

The mixture model reports this data instance of day 4 as an outlier due to its large distance to the assigned hidden state in Viterbi path, but there is no reasoning found by the single model indicators explaining why this point is detected as an outlier. As the table results show, none of the single indicators support this decision. However, based on ensemble learning [119] the mentioned data point is very likely to be an outlier as the majority of detectors acknowledge that. The single model, in this case, is not good enough to justify its detection, and the separate model does not recognize it at all. Several fluctuations in the separate model makes it impossible to distinguish any outlier for this example. It is only the mixture model which detects the point and has support from the HMM indicators.

Furthermore, there is another outlier suggested by the mixture model in day5 at 9:00 which is supported by two of the HMM indicators, the low probability and the rare transition between hidden states. Among the STOA detectors, the multivariate technique also marks this point as outlier. Neither the single nor the separate HMM models are able to detect this point. Both data points detected by mixture model are

TABLE 4.2: HMM Outliers Compliance with STOA Outliers and HMM Indicators

	STOA Outliers			HMM Indicators		
	Univariate	Multivariate	Temporal	Large Distant (Dist)	Low Probability (Prob)	Rare Transition (Trans)
Separate Model	29%	70%	74%	15%	9%	10%
Single Model	29%	60%	73%	22%	8%	24%
Mixture Model	21%	81%	76%	24%	12%	15%

compatible with the STOA outliers (one with univariate and both with multivariate), and they are both supported by the HMM indicators of the mixture model.

If the outliers can not be justified by the HMM indicators they are referred to as *soft false positives*. If they can not be explained neither by the HMM indicators, nor by the STOA outliers, they are declared as *hard false positives*. The unique outlier introduced by the temporal method is not detected by any of the HMM variations. The overall compatibility of the HMM outliers with the STOA outliers and HMM supported indicators are summarized in Table 4.2 and 4.3 in the next section.

The quality of HMMs in modeling observation sequences is an important concern. How accurately does an HMM represent the characteristics of its associated observations and adjust to the overfitting issues at the same time? Basically, each HMM model produces a number of false positives which in the absence of the precise ground truth is complicated to distinguish. In spite of this, a background knowledge of the network abnormalities in addition to the HMM parameters inspection provides reliable explanations to estimate anomalies and can improve the robustness of anomaly detection.

The Systematic Approach

In this section we present the likelihood results of three HMM variations for all the APs in the one week data set, comparing the detected outliers to STOA outliers and the HMM parameters indicators. Those HMM outliers not in agreement with any of the HMM indicators, are marked as *Soft FP*. If they are not compatible with HMM indicators nor with the STOA outliers they are referred to as *Hard FP* that means we found no reason for their selection as outliers.

This experiment also reveals the best HMM models for different types of anomalies. For instance Table 4.2 shows that mixture of HMM is the best model to capture multivariate outliers, while in detecting other types of the STOA outliers all three types of HMMs perform almost identical. The mixture model is more capable of recognizing outliers caused by large distances to the proper hidden state, while the single model is the most efficient model for outliers due to rare state transitions and large distances distinguished in Viterbi path.

Table 4.3 shows that more than 77% of the outliers detected by the three versions of HMM are in accordance with the STOA outliers, while the second column displays that around half of the HMM outliers can be validated by HMM indicators (large distance, low probability and rare transition). The higher the compliance between HMM outliers and HMM indicators, the more confidence is assured in the presented anomalies to the network administrating team. The single and mixture models demonstrate a superior rate of compliance to HMM indicators, while the ratio of the *Hard FP* in the mixture model still remains lower than the two other HMM models. It must be noticed that STOA outliers are not utilized as ground truth, but just as auxiliary verification tools providing an extra level of certainty for explaining the HMM outliers.

TABLE 4.3: Comparison of HMM Variations

	STOA	HMM Ind.	Soft FP	Hard FP
Separate Model	81%	45%	55%	12%
Single Model	77%	57%	43%	15%
Mixture Model	85%	56%	44%	6%

TABLE 4.4: Anomalous Patterns Observed in 3 HMM Variations

		AP Halt/Crash			Persistence Interference		AP Overload		
		210000	110000	220000	10 (3x)	20 (3x)	0202	0101	1212
Separate Model	AP (%)	10	25	12	14	0	1	18	8
	Day (%)	2.2	6.4	2.4	3	0	0.2	4	2.4
	Outlier Compliance (%)	10	20	33.3	64.2	0	0	5.5	12.5
Single Model	AP (%)	2	40	2	3	0	0	4	1
	Day (%)	0.4	13	0.6	0.6	0	0	0.8	0.2
	Outlier Compliance (%)	100	52.5	100	0	0	0	25	0
Mixture Model	AP (%)	4	31	7	9	1	0	10	1
	Day (%)	0.8	8	1.6	2.2	0.2	0	2.4	0.2
	Outlier Compliance (%)	25	22.5	28.5	33.3	0	0	40	0

4.5.2 Anomalous Patterns

Table 4.4 presents the observed anomalous patterns in separate, single and mixture HMM models by the percentage of the patterns' occurred per AP and per day, in addition to the patterns' compliance to the observed outliers. The *Outlier Compliance* row of this table shows in what portion of the observed anomalous patterns an outlier is also detected by the model.

Among the three categories of anomalous patterns, the least frequent one observed is *Persistence Interference* which is the repetition of downturns to the lowest state (*10* or *20*) more than three times a day. Generally the state transition of *20* is very rare and its three recurrences per day, is only captured once by the mixture model (in 1 hour of 1 day). However, transition from the medium state to the low state (*10*) is more frequently occurred and is identified by the separate model.

In the case of *AP Halt/Crash*, *110000* pattern appears more than the two other patterns (*210000* and *220000*). However the most salient patterns of this category is *220000*, which demonstrate a sudden decline of usage from peak to bottom that lasts for 4 consecutive time slots. It should be noted that such types of anomalies need a further check by the network administrator. For example if this pattern happens in classroom or auditorium, it can be due to the termination of a crowded event after which a large number of participants decide to disconnect their wireless stations and leave the place almost concurrently. Another interesting outcome is inspecting the pattern observations by various HMM models. For example when the single HMM detects (*220000*) pattern, it is extremely acceptable to be a true detection, while detecting (*110000*) pattern by the same model could contain many false alarms. The reason behind that is the high probability of watching only 2 states in Viterbi path (*0* and *1* or *1* and *2*) created by the single model, due to the HMM state generalization (expanded HMM states generated from the entire data set). The high rate of observing *110000* pattern by the single model (in 40% of APs and in 13% of days) and low rate of *220000* pattern observation (in 2% of APs and in 0.6% of days) supports the above discussion.

The *AP Overload* pattern which is identified by the HMM state fluctuations, is very unlikely between the high and low states (*0202*) and is captured just once appeared in the separate model. The (*1212*) pattern seems to occur less than *0101* pattern, and the former is a more appropriate estimation of *AP Overload*, as the extremely jammed hours (high HMM state) normally are followed by medium states in AP overloaded situations and the intermittent fluctuation between these two states can be a typical form of this type of anomaly.

Table 4.4 performs a cross-check between the distinguished anomalous patterns and the detected outliers of each HMM model. For example in the mixture model, the second pattern of the *AP Overload* (*0101*) is observed in 10% of the APs, which in 4 of them (40%), an HMM outlier has also occurred during the pattern at the specified anomaly point. For each anomalous pattern there is an anomaly point (HMM state in Viterbi path), in which the AP is crashed, the interference occurs, or the AP is overloaded at that specific state. For instance in the *AP Halt/Crash* pattern, the third state is where the anomaly happens (downturn to the low HMM state). Observing the previous and next hours just determines the type of anomaly and affirms the required duration. The compliance between the anomalous patterns and the observed outliers of each model provides a higher level of confidence for the proposed anomalies to the network management team.

The comprehensive analysis of such anomaly-related patterns or even more complicated alternatives, may inform the network managers about various types of anomalies, the level of severity, and the extent of confidence to the presented patterns. Such information may permit the network managers to take immediate actions or make long-term decisions for the maintenance or re-structure plans of the network.

4.6 Conclusion

The main contribution of this section was analyzing the user's behavioral patterns and learning models to detect anomalous patterns. We proposed a new application of HMMs in performance anomaly detection of 802.11 wireless networks and presented several indicators of outliers by HMM parameters' analysis. Furthermore, we provided a number of anomalous patterns associated with such networks in terms of HMM state transitions.

The experimental results show that HMM models were able to discover a portion of the state of the art's outliers (univariate, multivariate and temporal). The HMM models also introduced some additional outliers that could be justified by HMM parameter indicators (large distance to assigned HMM state, low transition probability, and rare state modification). The reason for evaluation through STOA outliers and HMM indicators is the lack of ground truth in the RADIUS data set. The single and mixture models outperformed the separate HMMs in terms of accuracy and HMM indicators conformity.

Chapter 5

Hidden Markov Model Analysis for Anomaly Detection: Time Variant Modeling

5.1 Introduction

Due to unstable radio conditions, faulty equipment, and dynamic user behavior among other reasons, there are always unpredictable connectivity problems in a wireless covered area. Detection and prediction of such problems is of great significance to network managers if they are to alleviate the connectivity issues of the mobile users and provide a higher quality wireless service. This chapter aims to improve the management of the 802.11 wireless networks by characterizing and modeling wireless usage patterns in a set of anomalous scenarios that can occur in such networks. This chapter contains two main divisions: time-variant and time-invariant HMM modeling in section 5.2, and Hidden Markov Models and Universal Background Model in section 5.3.

In section 5.2 we apply time-invariant (Gaussian Mixture Models) and time-variant (Hidden Markov Models) modeling approaches to a data set generated from the large production network addressed in section 3.2. In subsection 5.2.3 we propose to apply these models for anomaly detection purposes. We then evaluate the proposed anomaly detection methodologies in a controlled environment of the Testbed network addressed in section 3.3. In subsection 5.2.4 we analyze and discuss the experimental results of the Testbed showing that HMM outperforms GMM and yields a higher anomaly detection ratio and a lower false alarm rate.

In section 5.3 we propose an anomaly detection technique based on Hidden Markov Model (HMM) and Universal Background Model (UBM) on data that is inexpensive to obtain. Furthermore in subsection 5.3.2 we present techniques for detection of anomalous time-series in a database of time-series, distinction of anomalous patterns, and detection of anomalous points within a given time-series. We then evaluate our proposed methodologies on generated network anomalous scenarios in OMNeT++/INET network simulator addressed in section 3.4, and compare the detection outcomes with those in baseline approaches - RawData and Principal Component Analysis (PCA). The experimental results in subsection 5.3.3 show the superiority of HMM and HMM-UBM models in detection precision and sensitivity. In subsection 5.3.4 we present the summary and conclusion of the main section.

5.2 Time Variant HMM Modeling

5.2.1 Background

In this work, we propose an automatic diagnostic tool that analyzes the usage data of the APs—collected from a RADIUS authentication server. We apply probabilistic learning algorithms to produce a model for each access point or group of access points, and identify anomalous events with a margin of certitude. AP usage modeling and anomaly detection in hotspots would assist network administrators to ensure long-term quality of service by analyzing various connectivity factors of wireless users in particular localities. In the current work we focus on proposing individual models for APs as the ground truth data is only available through the single AP Testbed deployment. The prospective methodology is based on the development of HMM models and a detection tool using Wi-Fi campus data; other work that we have developed in [13] and [12] has taken this approach into account. As a preliminary investigation on the subject, we focused on short 802.11 sessions recorded through RADIUS authentication as a network artifact and an indicator of quality of wireless access in [13]. In [12] an exhaustive analysis is performed for outlier detection in 802.11 wireless networks using HMM variations- single HMM, mixture of HMMs and individual HMMs- and is evaluated by the state of the art statistical methodologies. Furthermore a number of network anomalous patterns are represented, in the same study, considering HMM parameters such as hidden states' transition and partial likelihood of the observation sequences. In the current study we consider Hidden Markov Model and its counterpart time-invariant methodology - Gaussian Mixture Model (GMM) - to investigate the temporal relevance of the employed data. These two methodologies are analyzed and compared with each other both in modeling and anomaly detection experiments. The key research question is whether time plays a role in the modeling or a simple time-invariant model such as GMM is adequate.

This section contains two main parts: 1) analysis and modeling of 802.11 AP usage and exploring the time dependency of the employed data, and 2) identification, detection, and characterization of different types of anomalies. The aforementioned objectives are investigated as case studies on the large data set of AP usage addressed in Section 3.2. Moreover those objectives are examined on the smaller scale Testbed defined in Section 3.3 for the purpose of evaluation.

5.2.2 Methodologies

In this section we introduce statistical techniques for modeling purposes and in the upcoming section we indicate how to apply these models for anomaly detection. Although in this thesis we use the modeling approaches for anomaly detection, they can be used in distinct directions such as investigating the similarities and differences of the locations, categorizing the localities in terms of functionality (e.g. classroom, office, library) or specification (homogeneous/heterogeneous daily, seasonal or constant usage). We introduce time-invariant and time-variant models and in each case we show how to apply the model on the large data set previously elaborated in Section 3.2.

Time Invariant Modeling: Gaussian Mixture Model

We first consider models that assume there is no time binding between consecutive daily events. Although this might not be precisely the case, it yields a modeling

approach that time does not play a role in it. Later in this study we compare this type of modeling with others that do consider dependency between consecutive daily events.

We begin our modeling efforts by applying techniques that assume all daily events come from the same distribution, regardless of any time dependency between the consecutive records. To explain this, we pick Gaussian Mixture Model (GMM), a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distribution with unknown parameters. The Expectation Maximization (EM) procedure is an optimization technique utilized to fit the unknown parameters and incorporate information about the covariance structure of the data as well as the centers of the latent Gaussians [120]. GMM can be thought of as a single-state HMM with a Gaussian mixture observation density, or an ergodic Gaussian observation HMM with fixed, equal transition probabilities [121].

A Gaussian mixture model is a weighted sum of M component Gaussian densities as given by Equation 5.1.

$$p(x|\lambda) = \sum_{k=1}^{M} \omega_k \, g(x|\mu_k, \Sigma_k) \tag{5.1}$$

where x is a D-dimensional continuous-valued data vector (of features), $w_k, k = 1, ..., M$, are the mixture weights, and $g(x|\mu_k, \Sigma_k), k = 1, ..., M$, are the component Gaussian densities. Each component density is a D-variate Gaussian function of the following form:

$$g(x|\mu_k, \Sigma_k) = \frac{exp\{-\frac{1}{2}(x-\mu_k)'\Sigma_k^{-1}(x-\mu_k)\}}{(2\pi)^{D/2}|\Sigma_k|^{1/2}} \tag{5.2}$$

with mean vector μ_k and covariance matrix Σ_k .The mixture weights satisfy the constraint that $\sum_{k=1}^{M} \omega_k = 1$.

The complete Gaussian mixture model is parameterized by the mean vectors, covariance matrices and mixture weights from all component densities. These parameters are collectively represented by the following notation:

$$\lambda = \{\omega_k, \mu_k, \Sigma_k\} \qquad k = 1, ..., M \tag{5.3}$$

GMM Application: Case Study GMM could be applied to our data features in several ways, for instance a single mixture model for the entire set of data, or a mixture model for each location separately. The later approach is closer to the goal of proposing practical models for each place indicated by an AP to explore the characteristics of that place, and ultimately discovering the abnormal behaviors occurring in contrast with the expected usage pattern.

In order to investigate the modeling capacities of GMM, we select two different spots to be our test cases: a highly crowded AP at a computer service section with 3726 observed users, and a less crowded AP in a chemical engineering department with overall 175 users. The experiment takes into consideration the second semester period of 2011 from February to July. To achieve more precise results, we focus on the working daily pattern, hence the data records belong to the working days (from Monday to Friday) and the working hours (8:00 to 18:00).

On each location, GMM fits are computed with three mixture components. The Gaussian density parameters (mean and covariance matrix) are depicted in Figure 5.1, the first row belongs to the crowded AP and the second row shows the density parameters of the less crowded AP. In order to facilitate the visual perception and

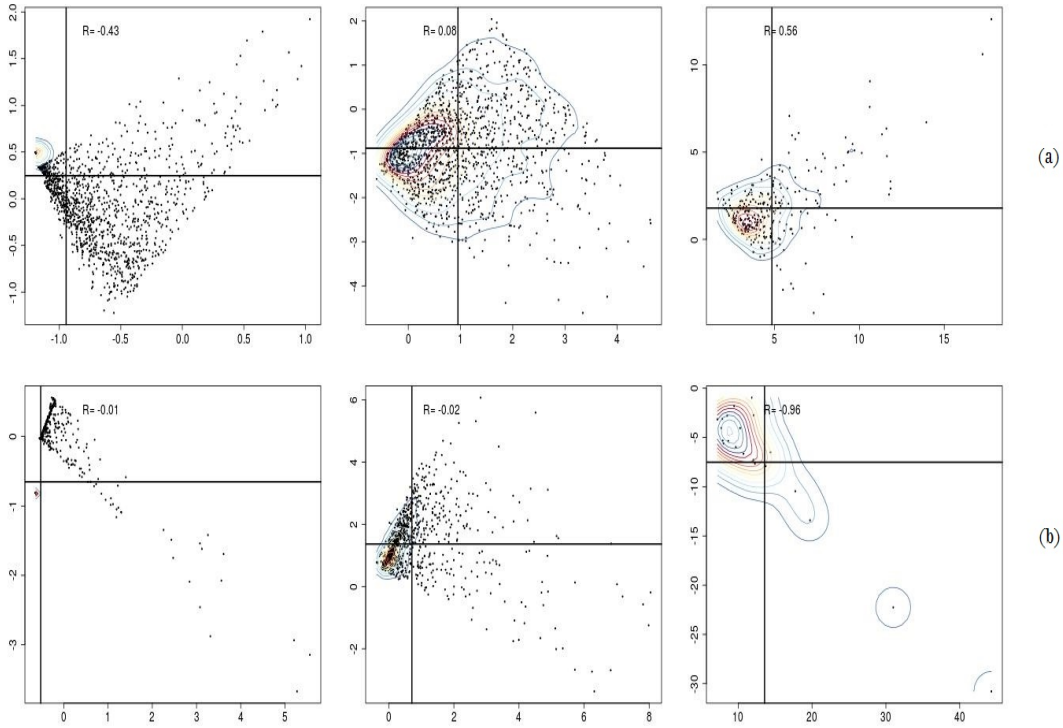


FIGURE 5.1: Density parameters of three Gaussian mixture components of the selected APs. **(a)** Crowded AP, **(b)** less crowded AP

to have an easier comparison, the density parameters are illustrated in 2D, despite the fact that GMM process is conducted on 3 principal components as explained in Chapter 3 subsection 3.2.3.

The data is standardized to have zero mean and one standard deviation feature-wise, so the density values are not appropriate to be compared with each other directly. However, the contour lines show the diversity of the data points in each mixture component and the direction of spread as well as the mass center. The R value on each plot represents the correlation between the X and Y axis, correspondingly the first two principal components. The distribution parameters of the GMM components shown in Figure 5.1 reveals that mixture components of the crowded AP model and less crowded AP model are not very much alike in terms of R values or direction of spread. However, the most and least intense mixture components can be observed in the first and the third components of both models, respectively.

Each location is characterized in this manner and according to GMM modeling approach represents the mixture weights and density parameters of the first and the second APs, respectively:

$\lambda_1 = \{\omega_{i1}, \mu_{i1}, \Sigma_{i1}\} \qquad i = 1, ..., 3$

and

$\lambda_2 = \{\omega_{j2}, \mu_{j2}, \Sigma_{j2}\} \qquad j = 1, ..., 3$

Time Variant Modeling: Hidden Markov Model

In this section we consider models that assume time dependency between consecutive daily events. In this case the sequences of data records matter and they form significant connections in a meaningful context or profile. In time-variant models

in general, conditional probabilities for events are determined based on the history of the events. In the following section we study the Hidden Markov Models for modeling the time-varying sequential data for the purpose of anomalous pattern recognition.

HMMs are generally used for the stochastic modeling of non-stationary time-series. HMMs provide a high level of flexibility for modeling and analyzing time-varying processes or sequential data. Their particular application is in recognition such as speech recognition, activity recognition, gene prediction, etc. where data instances are represented as a timely sequence of estimates. In the current research we propose to use HMMs for modeling and anomaly detection purposes in wireless networks which has never been investigated before to the best of our knowledge.

A more comprehensive definition of Hidden Markov Models are provided in Chapter 4 Section 4.3.1.

According to our data set, the HMMs form observations with continuous multivariate Gaussian distribution, hence the emission matrix B is defined by the distribution parameters associated with the set of states. In the proposed model, the HMMs contain fully connected states, thus transitions are allowed from any state to any other state.

HMM Application: Case Study In this section we select the very same APs as in the GMM case study (Section 5.2.2), and build HMM models for each of them separately. Our focus is once more on the working daily pattern in the second semester of 2011, from Monday to Friday in the working hours.

As described earlier, we consider fully connected HMMs (ergodic model) with continuous Gaussian distribution as the emission probabilities and 3 states. The states are initialized randomly, and we chose 3 states based on the best practice of the experiments conducted on both the large data set and the Testbed data set. For the multivariate Gaussian observations, the initial values of the mean vector are uniformly drawn between $\mu - 3\sigma$ and $\mu + 3\sigma$, and the initial variances of the diagonal covariance matrix are uniformly drawn between $\frac{1}{2}\sigma^2$ and $3\sigma^2$. The initial probability matrix (π) and the transition matrix (A) are uniformly drawn. The primary HMM is then optimized by means of the Baum-Welch algorithm with the cut off likelihood value of 1e$-$6 or the maximum number of iterations set to 20. Following the optimization process some states may better reflect usage or density given their values. For example a hidden state with the highest value for the second principal component shows a populated case in terms of users or sessions density.

The Gaussian density parameters of the three hidden states are illustrated in Figure 5.2. Similar to Figure 5.1, the first row is affiliated with the crowded AP and the second row belongs to the less crowded AP. The contour lines in these two figures represent the overall picture of the population and density distribution of the data in each GMM component or HMM state. Figure 5.2 shows that the first state of the crowded AP model and the less crowded AP model are very similar in terms of density of data points, direction of spread and R values. The second states of these models, however, have almost nothing in common. The second state of the less crowded AP model contains few data points and less intensity. The third states are similar in terms of direction of spread and data points density, however the R values showing the correlation between the first two principal components are quite different.

Suchlike graphs are visual aids to depict the distribution parameters, and for inspecting the goodness of fit over the entire feature set further analysis are required.

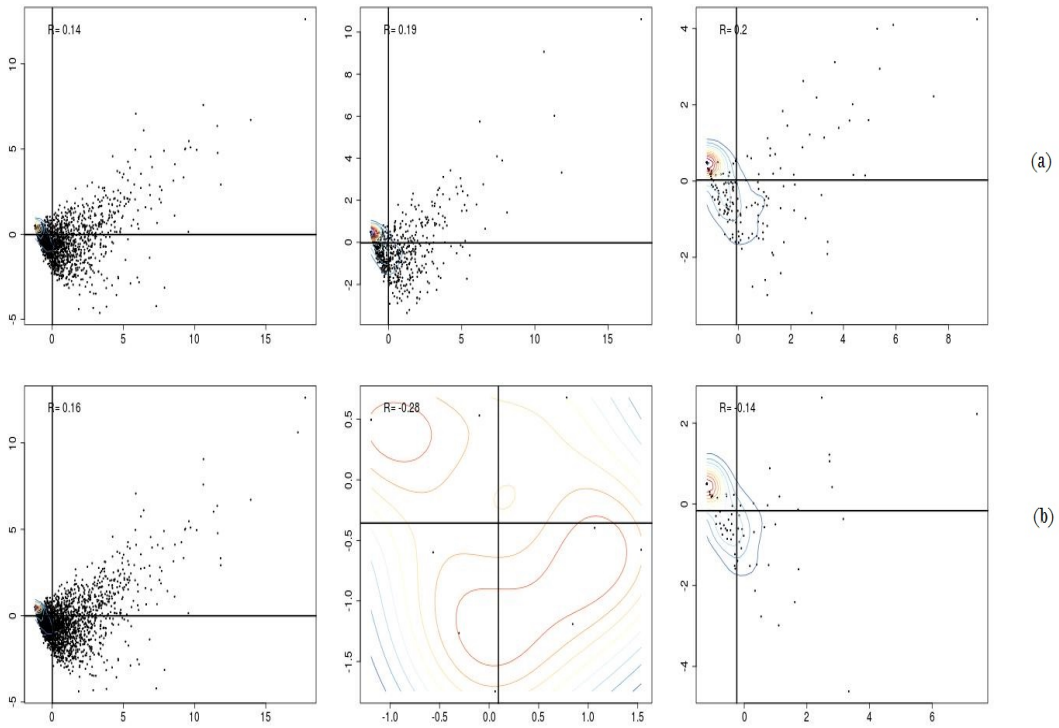


FIGURE 5.2: Density Parameters of Three Hidden State in HMM of the Selected APs. **(a)** Crowded AP, **(b)** Less Crowded AP

Model Comparison: GMM vs. HMM

In this section two techniques are considered only for the sake of modeling purposes, a time-invariant model (GMM) and a time-variant model (HMM). In the coming section we investigate the ultimate goal of this modeling which is the recognition of anomalous points or regions. At this stage, before exploring the anomaly detection territory, we briefly itemize the modeling functionalities and propose some simple tests to verify the more qualified model.

The potential functionalities of the locations characterization and modeling are listed as following:

- Classification of the locations, represented by APs, in terms of utility and temporal patterns.
- Recognition of the similarities and distinction of the locations.
- Grouping the most related APs and propose mixture models for the groups [12].

To investigate the competency of the two proposed models and estimate the capacity of each, we conduct a simple test. First of all, we measure the log-likelihood of the models in modeling the training data of the two samples, crowded AP and less crowded AP, and then we select a random day from each AP and calculate the log-likelihood of the models towards the test data which is new to both models. We use log-likelihood values (LLV) to measure the goodness of fit of our models. The model with larger log-likelihood value surpasses the model with smaller log-likelihood value.

Given data x with independent multivariate observations $x_1, ..., x_n$, the likelihood of a Gaussian mixture model with M components is defined as [122]:

$$likelihood(x|\lambda) = \prod_{i=1}^{n} \sum_{k=1}^{M} \omega_k \, g(x_i|\mu_k, \Sigma_k) \tag{5.4}$$

where $g(x|\mu_k, \Sigma_k)$ is the kth component's Gaussian density, as already defined in Equation 5.2, and ω_k is the probability that an observation belongs to the kth component.

The log-likelihood function takes the following form:

$$\text{log-likelihood}(x|\lambda) = \sum_{i=1}^{n} log(\sum_{k=1}^{M} \omega_k \, g(x_i|\mu_k, \Sigma_k)) \tag{5.5}$$

In the EM process, the parameters of the GMM, λ, are estimated so that the likelihood of the GMM given the training data is maximized, using Maximum Likelihood Estimation (MLE). Ensuing several iterations, the MLE yields the likelihood of the GMM given the training data. We applied MClust R package [123] to fit the Gaussian mixture components and estimate the log-likelihood of the training and test data.

The likelihood of a HMM is basically the first key problem of HMMs stated earlier, the probability of an observation sequence given the model parameters:

$$\begin{aligned} P(O|\lambda) &= \sum_{all\,H} P(O|H,\lambda)P(H|\lambda) \\ &= \sum_{h_1,h_2,...h_T} \pi_{h_1} b_{h_1}(O_1) a_{h_1,h_2} b_{h_2}(O_2) ... a_{h_{T-1},h_T} b_{h_T}(O_T) \end{aligned} \tag{5.6}$$

We utilized GHMM library in Python [124] for the formation of HMMs and estimation of log-likelihoods.

Table 5.1 contains the log-likelihood values of the trained GMM and HMM models for the selected APs, regarding the training and test data. Comparing the log-likelihood values of the training data, HMM provides higher values (less negative) both for the crowded AP and the less crowded AP. Note that the training data contains 25 days data and the test data consists of only one day data selected randomly from the unobserved days. Concerning the test data, it is expected that the selected day from the same AP obtains higher log-likelihood value rather than the data from another AP due to the possible similarity of daily usage in a specified location. The first GMM (built over the crowded AP data) provides the same amount of log-likelihood for both test data, thus yields no distinction between its own usage pattern and the other AP. However, the second GMM (trained with the less crowded AP data) provides higher log-likelihood value for its own data rather than the other AP.

TABLE 5.1: Log-likelihood Values (LLVs) of the Training and Test Data Belong to the Selected APs for GMM and HMM Models

Trained Model / Test Data LLVs	GMM Crowded AP	GMM Less Crowded AP	HMM Crowded AP	HMM Less Crowded AP
The same train data	-3468	-2154	-2553	-2131
Test data from the crowded AP	-189	-189	-134	-209
Test data from the less crowded AP	-509	-95	-195	-115

HMMs, on the other hand, provide higher log-likelihood value for their own test data than for the other AP, which shows the better matched model for self data. It must be considered that the test data is selected randomly and the pattern of the selected day is not determined in terms of normal or abnormal usage. In Section 5.2.4, the experiments are conducted on a Testbed data set with recognized anomalies so that the conclusion will be based on the known ground truth. In the next section, we investigate the time-variant specifications of HMMs towards the simplicity of the time-independent GMM concerning the anomaly detection objectives.

5.2.3 Anomaly Detection in AP Usage Data

Network administrators are generally concerned about anomaly detection as well as prediction. These two important tasks enable them not only to make immediate decisions to alleviate the complications of the network, but also to establish long-standing plans to support the expansion of the network and its dynamic usage over time.

In this section we show how the aforementioned models in Section 5.2.2 are utilized for the purpose of anomaly detection.

GMM Estimation: Divergence from Gaussian Densities

The most generic definition of the anomalies asserts those points or small regions isolated from the normal zones which contain the majority of the observations. Thus, a straightforward approach to detect anomalies, when there is no ground truth available, is to define the normal zones and distinguish those rare observations which hardly belong to those normal zones.

In the GMM model discussed earlier, a number of Gaussian mixture components are determined and each component contains normal density parameters. The model is built based on several training data and the newly arrived records are inclined to the most compatible component with the least distance. Hence, to detect abnormal points we need to estimate the affinity degree of each point, as already described in Equation 5.4 and 5.5, and mark outliers as having the slightest probability of belonging to any cluster.

HMM Estimation: Likelihood Series

HMM, as a time-variant model, considers the temporal dependency between consecutive data records. Calculating the log-likelihood of a single data point or a series of sequential data points as already expressed in Equation 5.6, emanates the mis-behaving records comparing to the log-likelihoods of the norm of the data. The unexpected low values of the log-likelihood in HMM are generally due to: 1) large distance from the assigned hidden state, 2) less likely state transition, or 3) hidden states' unbalanced separation, explained earlier in Section 4.4.1.

Anomaly Detection: Case Study

We explore the addressed methodologies to detect anomalous data points or data sequences in the same two APs that GMM and HMM models were trained for them. Figure 5.3 highlights the outliers detected by measuring the largest distance from the Gaussian components. The data used in this experiment belongs to the same test day of the previous section. The result of the first AP (crowded AP) is displayed in blue

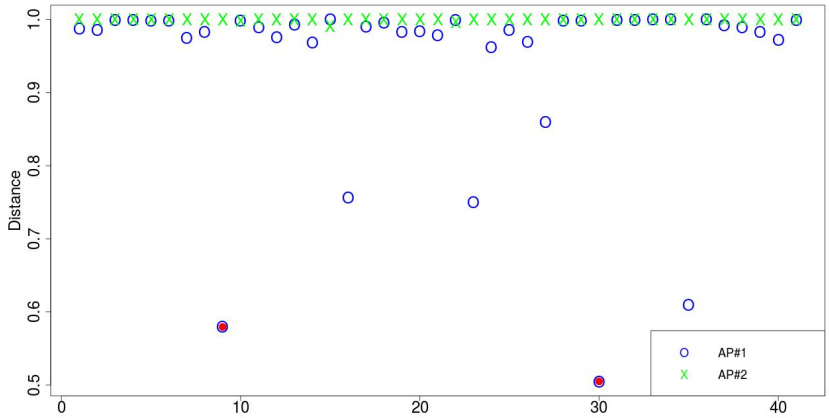


FIGURE 5.3: GMM Estimation of Anomalous Data Points Based on the Largest Distance from the Assigned Gaussian Component

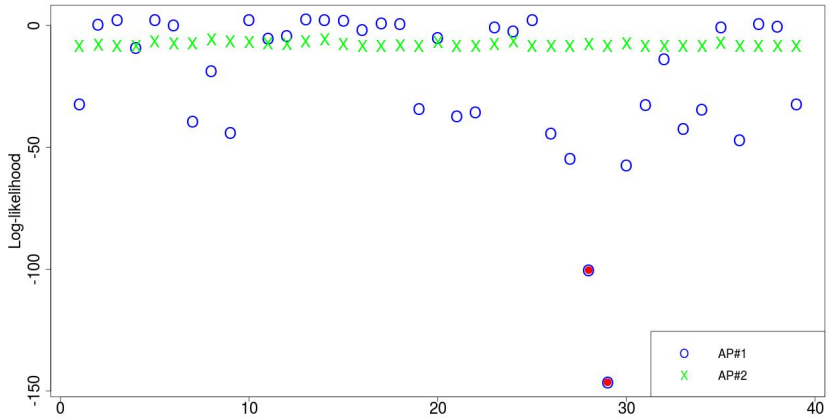


FIGURE 5.4: HMM Estimation of Anomalous Data Points Based on the Lowest Log-likelihood

circles and the second AP (less crowded AP) is demonstrated in green Xs. Two detected outliers are marked in red that both belong to the first model of the crowded AP. These outliers are appointed to a Gaussian component of the first model, but with the lowest probability (less than 60%). Here we selected the normality threshold to be 60%, however it could differ from model to model and the most appropriate value of threshold could eventually be tuned by the network manager.

Figure 5.4 displays the anomalous points detected by HMM based on the lowest value of the log-likelihood. In this approach, two different data points are marked as outliers which belong to the first AP training data, the crowded AP. The cut-off value is considered to be log-likelihoods below -100, note that this value could also be configured. More strict cut-off value should yield higher false positive rate. We investigated the likely origins of the outliers emerged in this case and we observed that the *Mahalanobis* distance of the marked data points are maximal with the assigned hidden state in the *Viterbi* path. That must have caused the low log-likelihood value in the likelihood series. No outliers were detected for the second AP (less crowded AP) neither by GMM nor by HMM.

In this case study we demonstrated how the anomaly detection analysis work in our proposed framework. In the next section, we evaluate both models based on the

achieved results of the deployed Testbed 3.3, as we can determine which points are detected correctly.

5.2.4 Experimental Results: Testbed Deployment

In order to validate anomaly detection techniques proposed in this work we deployed an exploratory Testbed with one single AP and generated a number of anomalies in a controlled environment for experimental purposes. More technical details on this data set can be found in Section 3.3.

In the following paragraphs we show how the modeling and anomaly detection techniques operate in the presence of the ground truth - data obtained from the Testbed deployment.

GMM vs. HMM Modeling: Pros and Cons

For the first experiment, a GMM model is built with 10 randomly selected normal days as training data. From then on, the likelihood of the generated model is computed against the training data as well as 10 unobserved normal days and 10 abnormal days as test data. The same process is performed on the HMM model, with the same set of training and test data. The summary of this experiment is displayed in Figure 5.5 and 5.6.

Both figures demonstrate overall higher likelihood values for the training data. The likelihood values of the unobserved data set is divided into normal and abnormal outputs which are displayed in graphs with different colors and shapes. In both models there are higher likelihood values for the normal days rather than the abnormal days. However, there is a discernible boundary between the normal and abnormal results in HMM while in GMM the likelihood values are not clearly separated and there are even some instances that the likelihood value of the normal day is lower than the abnormal day. The daily likelihood of abnormal days are apparently lower than the normal days, and this value varies with the number of abnormal occurrences and duration of each event. However, it is more straightforward to define a threshold for HMM rather than GMM model, to decide if a day is normal or abnormal.

Anomaly Detection

In this section we determine the anomalous time-slots with the proposed methodologies and compare the achieved results from the model with the Testbed anomalous ranges recorded for the abnormal instances. Note that various thresholds for each technique produce different results as the detection and false positive rates change based on the selected threshold. We use some statistical metrics to measure the detection accuracy and false alarms such as fall-out or false positive rate (FPR), specificity (SPC) or true negative rate (TNR), sensitivity or true positive rate (TPR), and eventually accuracy (ACC) and F1 score. We follow the definitions in [125].

The summary of the analysis on the normal and anomalous test data are presented in Table 5.2 for GMM modeling and in Table 5.3 for HMM modeling approaches.

Table 5.2 shows that higher thresholds increase the possibility of anomaly detection (24.9% rather than 4.7%), however the false positive rates also increase accordingly (19% rather than 9.9% and 3%). In normal test data, when we expect no

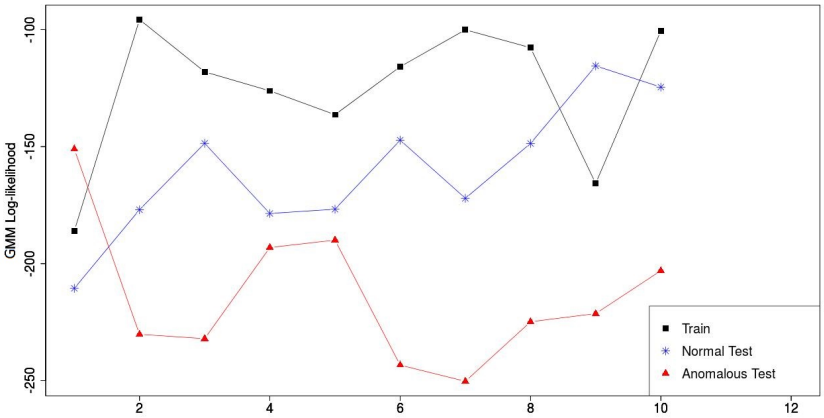


FIGURE 5.5: Likelihood values of the training and test data belong to Testbed for GMM Model

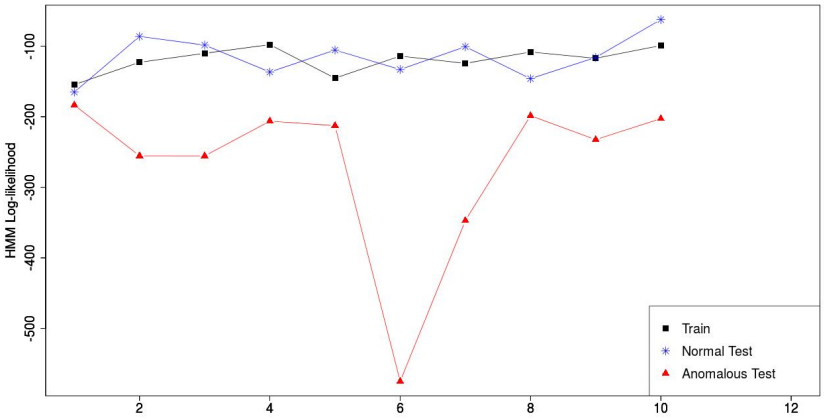


FIGURE 5.6: Likelihood values of the training and test data belong to Testbed for HMM model

TABLE 5.2: Anomaly detection of the normal and anomalous test data belong to Testbed for GMM

Statistical Metrics / Data - Threshold	FPR	TNR	TPR	ACC	F1 Score
Normal Testset (Threshold: 0.6)	2.5%	97.5%	0%	97.5%	0%
Normal Testset (Threshold: 0.7)	5.5%	94.5%	0%	94.5%	0%
Normal Testset (Threshold: 0.8)	10.5%	89.5%	0%	89.5%	0%
Anomalous Testset (Threshold: 0.6)	3%	97%	4.7%	81%	8.1%
Anomalous Testset (Threshold: 0.7)	9.9%	90.1%	4.7%	75%	7.2%
Anomalous Testset (Threshold: 0.8)	19%	81%	24.9%	70%	20.75%

TABLE 5.3: Anomaly detection of the normal and anomalous test data belong to Testbed for HMM

Statistical Metrics / Data - Threshold	FPR	TNR	TPR	ACC	F1 Score
Normal Testset (Threshold: -50)	0.5%	99.5%	0%	99.5%	0%
Normal Testset (Threshold: -20)	1.75%	98.25%	0%	98%	0%
Normal Testset (Threshold: -10)	3.75%	96.25%	0%	96%	0%
Anomalous Testset (Threshold: -50)	0%	100%	39%	90%	49%
Anomalous Testset (Threshold: -20)	0%	100%	43%	91%	52%
Anomalous Testset (Threshold: -10)	1.1%	98.9%	75%	95%	74%

anomalies to occur, from 2.5% to 10.5% fall-out is observed. Comparing this fall-out ratio to the results of Table 5.3 for normal test set, it is noted that much lower false alarms is marked for HMM (from 0.5% to 3.75%). Furthermore, the FPR for the anomalous data in HMM is quite small relative to GMM FPR output (1.1% in HMM vs. 19% in GMM). The highest detection rate or TPR in HMM modeling is achieved with Threshold equals to -10 which is 75% in average for 10 abnormal days of the experiment.

Regarding the FPR or fall-out ratio recorded for normal data in HMM, a careful consideration on each false alarm is performed and it is noted that the HMM model is slightly sensitive to extreme download ratio and in some cases both download and upload volumes. As the Testbed is deployed in a real home environment with real wireless users, although in normal days that no anomaly is generated deliberately, there might be some evidences of rather high download or upload by the users as it happens quite often in every wireless network. Therefore the false positive examples occurred in normal days could be introduced as real anomalies appearing in normal days, however for this experiment we assumed that normal days contain no anomalies.

TABLE 5.4: Detection rate of various anomalous patterns of the Testbed

Anomalous Patterns / Model	Jamming Channel (Short Intervals)	Jamming Channel (Long Intervals)	Heavy Usage (Single User)	Heavy Usage (Multiple Users)	AP Power Off
GMM (Threshold: 0.8)	28.5% (4/14)	17.3% (4/23)	8.3% (1/12)	0% (0/3)	35.2% (6/17)
HMM (Threshold: -10)	71.4% (10/14)	73.9% (17/23)	83.3% (10/12)	100% (3/3)	82.3% (14/17)

Table 5.4 displays the total proportion of different anomalies' occurrences in the Testbed and presents the detection rate of each anomalous pattern by GMM and HMM. Here we consider the anomalous test data and the highest likelihood thresholds of both models (0.8 for GMM and -10 for HMM) that provide the maximal detection rate. Detection ratio is determined by the overall number of time-slots marked as anomaly divided by the total number of time-slots.Comparing GMM and HMM once more demonstrates the superior capability of HMM in recognition of anomalous events, while providing unnoticeable false positive rate (Table 5.3). Among the

various types of anomalies generated for the Testbed, the highest detection rate belongs to heavy usage pattern, producing by multiple users and then single user. The lowest detection ratio, however, originates from jamming channel with short interval. Although there are some specific anomalous points that are never detected by the model, regardless of the cut-off threshold, the overall detection rate of the HMM is quite satisfactory.

5.2.5 Summary

Proposing time-invariant and time-variant modeling approaches and utilizing those models for anomaly detection in addition to a RADIUS Testbed deployment with simulated anomalies compose the key contributions of this work.

We presented GMM as time-invariant and HMM as time-variant modeling techniques. As a case study for each approach we selected two different locations in the university campus - a highly crowded AP and a less crowded AP - to apply the forenamed methodologies on them. We then defined the log-likelihood for each method separately to examine the goodness of fit for the proposed models in terms of train and test data. Having conducted a simple experiment on the selected APs revealed that HMMs are more likely to provide a robust model to distinguish between their own pattern and an unfamiliar pattern.

Furthermore, we described the anomaly detection techniques by GMM and HMM. In GMM we define anomalies as the distant data points that hardly belong to any Gaussian component, and in HMM anomalies are the data points with the minimum likelihood value. We analyzed the root cause of the low likelihood values in GMM as distance from the Gaussian components, and in HMM as divergence from the assigned hidden states as well as the low probability in state transition. We further explored the addressed methodologies to detect anomalies at the same two APs. We justified the detected anomalous points, however in absence of the ground truth in the large data set it was not possible to thoroughly evaluate the anomalous points, so we left the evaluation process for the Testbed experimental results.

We then applied our proposed models to detect anomalous points in an exploratory Testbed deployed in a home environment, and discussed the effectiveness of each model. We measured the false positive rate (FPR), true negative rate (TNR), true positive rate (TPR), accuracy (ACC) and F1 score in normal and anomalous test data. The experimental results demonstrated that HMM outperformed GMM in obtaining higher detection ratio while producing minor false alarm.

5.3 Improved Initialization Modeling and Anomaly Detection Results

5.3.1 Background

In the current study we improve our Hidden Markov Model (HMM) formerly proposed in Section 4.3 and 5.2 by integrating it with the concept of Universal Background Model (UBM). HMM-UBM is a large HMM trained to represent the AP-independent distribution of features by pooling plenty of data by the EM algorithm, or by pooling the sub-population models trained by individual UBMs [121]. The independency of the model applies both to APs and to anomalies, meaning that a large set of data from every AP is utilized for initialization regardless of containing anomalies or not. We apply UBM to initialize the HMM models using all the data available assuming general initial values provide more robust model estimates.

The simulation data previously addressed in Section 3.4 are utilized to evaluate HMM and HMM-UBM models and compare the anomaly detection results with baseline approaches (RawData and PCA).

The key steps of the present work include: 1) Conducting 802.11 wireless network simulation in OMNeT++/INET to resemble normal and anomalous scenarios. 2) Reiterating the simulations with different seeds to provide miscellaneous replicates. 3) Extracting the wireless users' data, and converting it to AP usage data. 4) Building HMM and HMM-UBM models from the prepared data set. 5) Applying the proposed anomaly detection algorithms. 6) Calculating the detection rate and sensitivity for evaluation purposes.

Regarding the anomaly detection techniques we analyze three main approaches: 1) detection of anomalous time-series in a database of time-series, 2) distinction of anomalous patterns, and 3) detection of anomalous points within a given time-series.

Furthermore, this study explores the following research questions: 1) whether HMM and HMM-UBM models are capable of anomaly detection and anomalous pattern recognition in AP usage data, 2) whether HMM and HMM-UBM models are required for anomaly detection or the baseline approaches are enough, 3) whether HMM-UBM have any advantages over HMM.

5.3.2 Methodologies

We use Hidden Markov Models adapted from a Universal Background Model for 1) detection of anomalous time-series, 2) distinction of anomalous patterns, and 3) detection of anomalies within a given time-series.

Universal Background Model

A universal background model (UBM) is a model used in a biometric verification system to represent general, person-independent feature characteristics to be compared against a model of person-specific feature characteristics when making an accept or reject decision. For example, in a speaker verification system, the UBM is a speaker-independent Gaussian mixture model (GMM) trained with speech samples from a large set of speakers to represent general speech characteristics. Using a speaker-specific GMM trained with speech samples from a particular enrolled speaker, a likelihood-ratio test for an unknown speech sample can be formed between the match score of the speaker-specific model and the UBM. The UBM may also be used while training the speaker-specific model by acting as the prior model in maximum a posteriori (MAP) parameter estimation [126].

We applied UBM to initialize the HMM models using the data available from all AP experiments regardless of containing anomalies or not. This is advantageous as in unsupervised learning approach the anomalous events are not known beforehand. Assuming that the HMM models adapted from a UBM produce as promising results as HMM models trained with normal data, achieving a qualified model even in the absence of the labeled data is more feasible. This in turn facilitates the process of unsupervised modeling. We later compare the detection results of the HMMs initialized with and without UBM in Section 5.3.3.

Given the data for training a UBM, there are many approaches that can be used to obtain the final model. The simplest is to merely pool all the data to train the UBM via the EM algorithm (Figure 5.7-a). One should be careful that the pooled data are balanced over the sub-populations within the data. Otherwise, the final

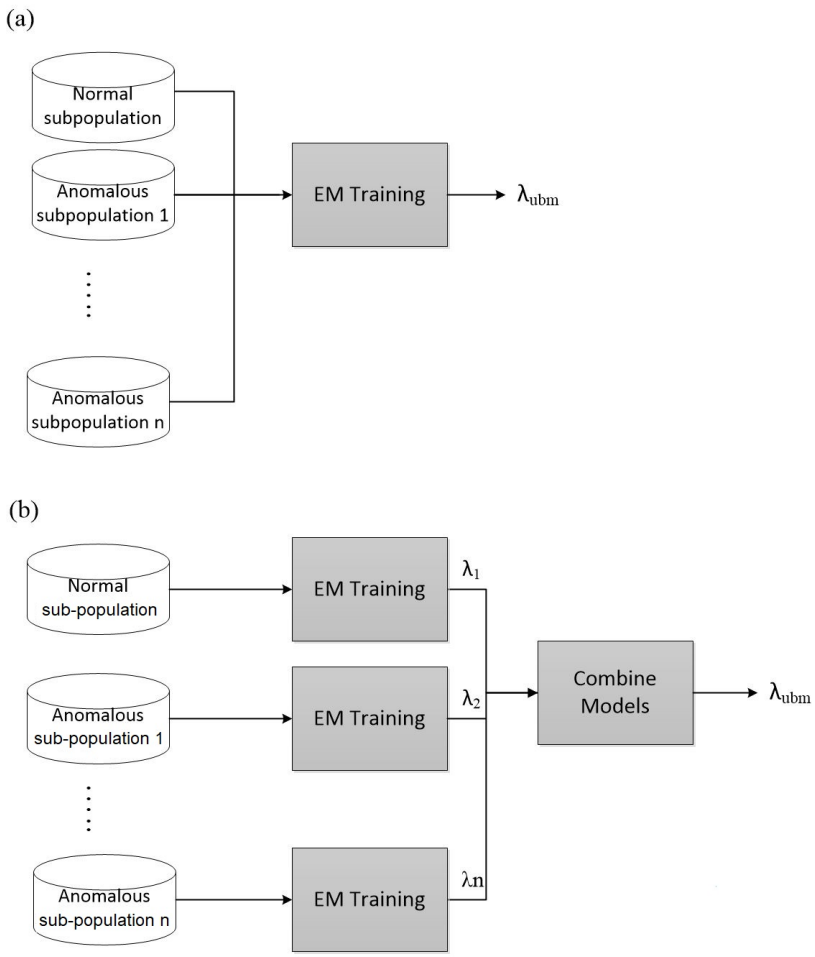


FIGURE 5.7: Data and model pooling approaches for creating a UBM. (a) Data from subpopulations pooled prior to training the final UBM. (b) Individual subpopulation models trained then combined to create final UBM.

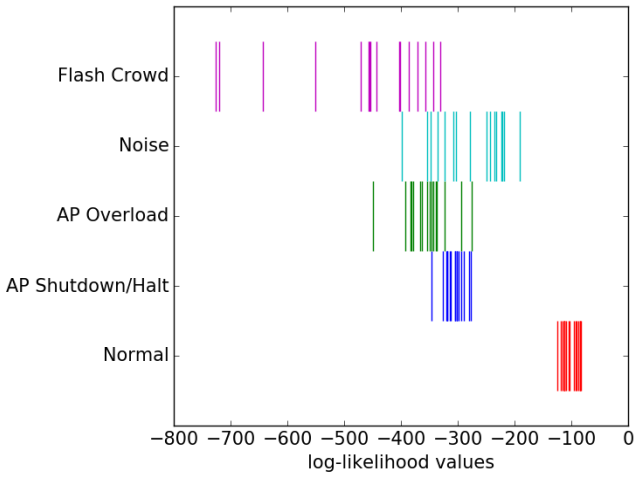


FIGURE 5.8: Log-likelihood values of normal and anomalous experiments.

model will be biased toward the dominant sub-population [127]. Another approach is to train individual UBMs over the sub-populations in the data, and then pool the sub-population models together (Figure 5.7-b). The latter approach has the advantages that one can effectively use unbalanced data and can carefully control the composition of the final UBM [127]. In our model we used the first approach, and to avoid a biased model we included the same amount of normal and anomalous data sequences. Half of the data set contains normal samples and the rest consist of anomalous events (equal portion for each anomaly).

Detection of Anomalous Time-Series

The goal of this type of anomaly detection is to find all anomalous time-series in a database of time-series, and to distinguish normal days from those that contain a number of anomalous events. Similar to traditional outlier[1] detection methods, the usual approach is to learn a model based on all the time-series in the database, and then compute an outlier score for each sequence with respect to the model [127]. In our case, we build an HMM model with UBM initialization using the training data of all the simulation experiments. Then we calculate the log-likelihood values of each time-series in the test data set. Those observation sequences that contain one or more anomalous events are expected to get lower log-likelihood values.

Figure 5.8 shows the range of the log-likelihood values belonging to the normal and anomalous experiments. The anomalous cases consist of *AP Shutdown/Halt*, *AP Overload*, *Noise*, and *Flash Crowd* scenarios. As this figure displays there is a distinction between the log-likelihood values of the normal cases and the rest of the anomalies. However, the anomalous cases are not completely separated and there is an overlap between them. The log-likelihood values of the AP Overload, Noise and AP Shutdown/Halt scenarios are approximately in a similar range. However, those of the Flash Crowd scenario are slightly lower than the rest and take a widespread range while the values of the AP Shutdown/Halt scenario are condensed in a limited range.

[1] We use *outlier* and *anomaly* interchangeably in this context.

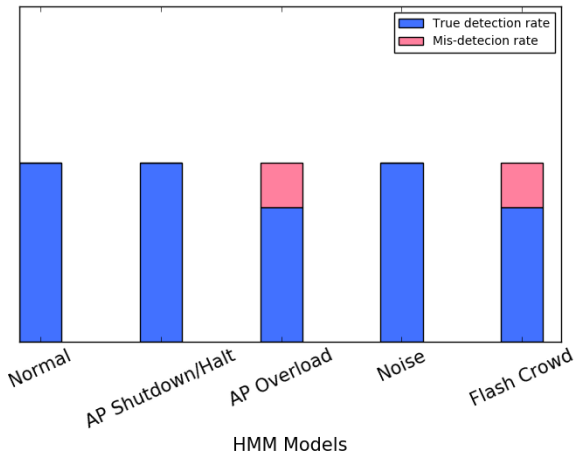


FIGURE 5.9: Detection results of the observations sequences by the trained HMM models.

As a conclusion, all the anomalous cases obtain log-likelihood values less than the normal range and thus it is feasible to distinguish the anomalous time-series from the normal ones. However, due to the overlapping log-likelihood values of the anomalies, it is not that simple to make a distinction between the anomalous scenarios just by inspecting their log-likelihood values. In the next section we consider modeling the anomalous cases independently to facilitate the distinction process.

Distinction of Anomalous Patterns

To capture distinctive characteristics of the anomalous scenarios we build separate HMM models for each anomalous scenario and also one model for the normal scenario from the training set. Then we compute the probability of each observation sequence from the test set getting generated by these models. The HMM model that produces the highest log-likelihood value is considered to be the generative model of the given time-series. At the end of this process we obtain a 2D matrix whose rows and columns consist of HMM models and log-likelihood of observation sequences, respectively. Choosing the best λ model among the competing models is termed as *scoring problem* and is a function of log-likelihood values.

Figure 5.9 presents the detection results of the HMM models given the normal and anomalous observation sequences from the test set. The x-axis contain the trained HMM models and the blue parts of the bars demonstrate the percentage of time-series correctly detected by their corresponding models. The top pieces of the bars in pink show the mis-detection ratio that occurs in AP Overload and Flash Crowd scenarios. 25% of time-series containing AP Overload anomaly are detected to be generated by Flash Crowd model. Moreover, 12.5% of Flash Crowd sequences are detected to be created by AP Overload model and 12.5% of them by Noise model. Besides these small mis-detection errors, the distinction process yields promising results in recognition of different anomalous patterns.

Each anomalous time-series in our experiment contains a single anomaly, while in reality each time-series can contain no anomaly (in normal cases) or various types of anomalies (in anomalous cases). A methodology to detect anomalous periods and distinguish between different anomalous patterns in unlabeled data is required to be performed in an unsupervised manner. Here we propose the basic scheme of

an algorithm which is based on the general model training in [128], and is adapted to our specific modeling approach and requisites:

1. A general HMM model is estimated with a large number of training samples (HMM-UBM).
2. Slice the first test sequence into fixed length segments. The segment(s) with the lowest log-likelihood given the general model in 1 is identified as anomaly.
3. A new anomalous model is adapted from the general model using the detected anomaly. A normal model is adapted from the general model using the other segments.
4. Slice the next test sequence into fixed length segments. Estimate the log-likelihood values of all segments given the previous adapted models (normal and anomalous models of step 3).
5. Update the adapted models using those segments that achieve closer log-likelihood to each model. Adapt a new anomalous model from the general model using any segment that achieves extremely low log-likelihood given the existing models (a new anomaly that hardly belong to any previous model).
6. Repeat step 4 and 5 until there is no more test sequences.

There are a number of parameters in this algorithm that is to be learned and determined, for example the length of the fixed-size segments, and the proper threshold for anomaly detection. However, by the end of this algorithm we expect to have one normal model and several anomalous models each presenting a specific anomalous pattern. Further post-processes are also applicable to merge the very similar models (by measuring models' distance) and yield the most optimized set of final models. More accurate explanation and implementation of this algorithm is out of the scope of the current study and is set aside for the future work.

Detection of Anomalous Points within a Given Time-Series

In this approach the anomaly score (log-likelihood) is computed for each data point given the trained HMM model. The unexpected low log-likelihood values show the divergence from the normal model and are typically indicative of anomalies. This method localizes the anomalous points or sub-sequences more precisely in the test sequence.

To detect the anomalous points in the log-likelihood series automatically, we propose a technique called *threshold detection* to define a boundary where the lower values belong to the anomalous set.

As many anomaly detection algorithms presume, outliers are the minority group not following the common pattern of the majorities. Accordingly we look for the extreme data points (outliers) with the lowest log-likelihood values. To this end a univariate histogram is constructed and the relative frequency (height of the histogram) is computed. The frequency of samples falling into each bin is used as an estimate of the density. We assume the samples with the highest density (mode) are the normal data points, and thus the bins containing the lowest frequencies and farther from the mode are the outliers. As a rule of thumb we mark bins with frequencies lower than a quarter of mode as outliers. Like any other change detection algorithm ours as well produce false positives, however in all the performed experiments of this work the false positive ratio is insignificant.

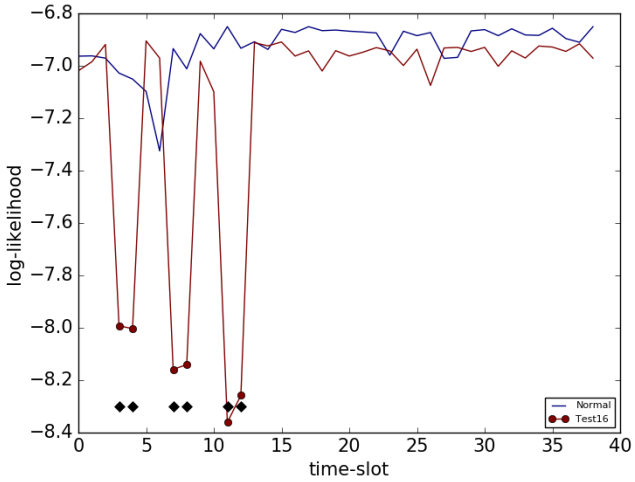


FIGURE 5.10: Log-likelihood of the normal model together with an example anomaly related to AP Overload experiment.

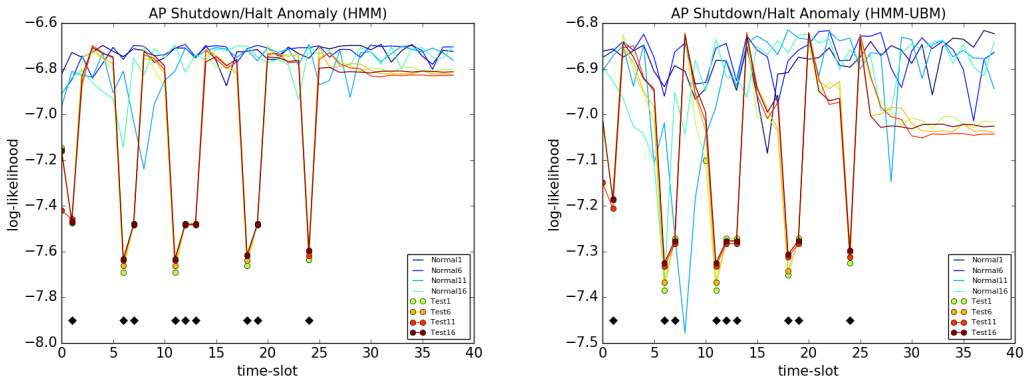
We use the same algorithm to detect the outliers or anomalies in RawData and PCA for the purpose of comparison. However, as RawData contains seven features, we conduct the algorithm on each single feature and aggregate the detected points as the final outcome. For example for the likelihood series of $s_1 s_2 ... s_{40}$, the algorithm detects s_2 and s_4 as outlier points for the first feature and s_4 and s_{15} for the third feature and for the rest of the features no anomaly is detected. In this case the final anomalous set contains $\{s_2, s_4, s_{15}\}$. The same method is applied to the PCA components to detect the anomalous points for three principal components.

Figure 5.10 demonstrates the log-likelihood values of an example anomalous case (AP Overload) generated by simulation. The red points are the anomalies detected by *threshold detection* algorithm and the black diamond points show the real anomalous period.

We further explore this type of anomaly detection in the following section and analyze each anomalous case specifically in more detail.

5.3.3 Experimental Results: Wireless Network Simulation

In this section we explore a set of simulated anomalous scenarios, present the HMM and HMM-UBM results for anomaly detection, and compare them to baseline approaches (RawData and PCA) for evaluation. In terms of HMMs, we consider fully connected models (ergodic), continuous observations with Gaussian distributions, and 3 hidden states. We believe that the HMMs with 2 states are too simple to capture the diverse characteristics of the locations (APs), while there is not enough variety in day-long sequential data for 4 or higher number of states. The simulations in this section are performed for 5 APs and 30 wireless stations that are basically a smaller version of simulations addressed in Section 3.4. Each experiment is repeated at least 20 times with different seeds in order to provide richer data set on slightly different samples. 80% of the data sequences are used for training the model and 20% is kept for testing.



(A) HMM (B) HMM-UBM

FIGURE 5.11: The log-likelihood series and detected anomalies of AP shutdown/halt scenario in HMM and HMM-UBM models.

AP Shutdown/Halt

When there is no session recorded for a given AP in RADIUS accounting table in a period of time, it is likely that the AP has stopped working - possibly due to a technical problem or power failure. In our simulation, we reproduced this anomaly by turning off the AP power deliberately during the *halt-period* for some *time-slots*.

Figure 5.11 demonstrates the HMM likelihood series and the anomalies detected for the test data set of this scenario. The valley shapes in this image shows the sudden drops of the likelihood values during the anomalous periods, and the marked points are the anomalies detected by the aforementioned *Threshold Detection* algorithm. The black diamonds show the actual anomalous points generated during the simulation.

Both HMM and HMM-UBM detect shutdown periods, even the short ones that only last one time-slot. However, Figure 5.11a shows that the HMM model built with only normal data gives a clearer model rather than the HMM-UBM model built with the entire data set including the anomalous cases in 5.11b. Despite this, HMM-UBM obtain adequate values for precision and recall, and even higher precision results in some cases.

Figure 5.12 shows the boxplot diagram of the anomaly detection's precision and recall computed for RawData, PCA, HMM, and HMM-UBM models. In these experiments both HMM and HMM-UBM achieve higher precision values and smaller false positive ratios compared to the baseline approaches (RawData and PCA).

Note that this type of anomaly is not very difficult to be detected just by looking at RawData as there is a visible change in data set features when the power is gone and no session is recorded. That is the reason RawData attains 100% recall. However, it produces relatively high false positive result that yields low precision.

AP Overload

In this anomalous case, excessive channel utilization occurs that could be the consequence of excessive download or upload by a number of wireless users. In this experiment we simulated AP heavy usage caused by all of the users of a given AP. Burst server (srvHostBurst) sends UDP packets to the given IP addresses in bursts

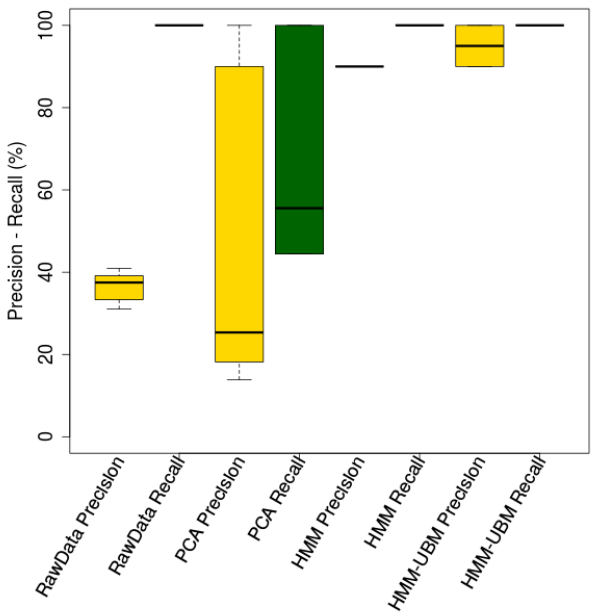
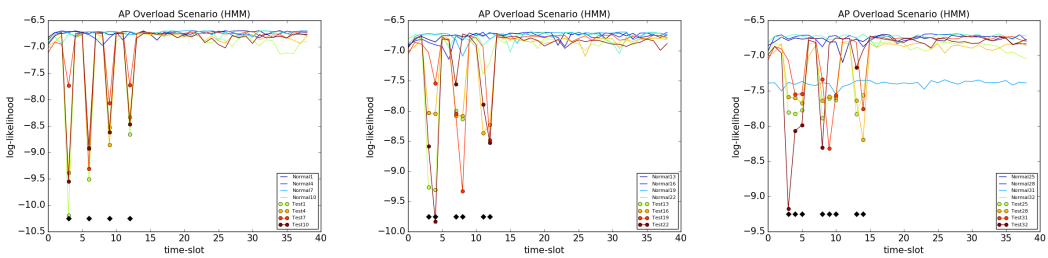


FIGURE 5.12: Precision and recall boxplot of RawData, PCA, HMM and HMM-UBM belong to AP shutdown/halt scenario.



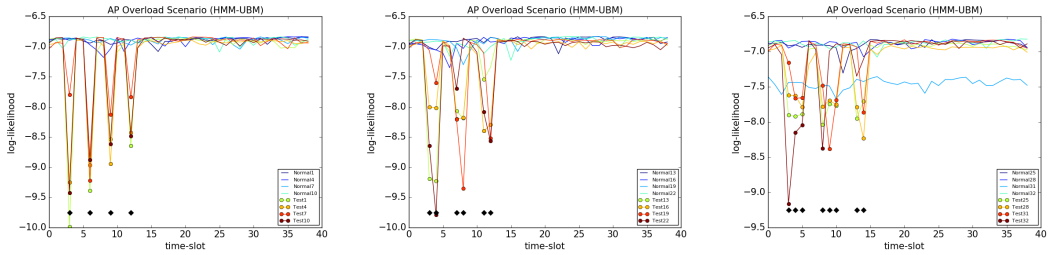
(A) burst-duration < sleep-duration (B) burst-duration = sleep-duration (C) burst-duration > sleep-duration

FIGURE 5.13: The log-likelihood series and detected anomalies of AP overload scenario (HMM).

during the *burst-duration* period which resembles the heavy downloads of the wireless users. In the *sleep-duration* period the burst flow stops and the channel utilization gets back to normal.

Figure 5.13 and 5.14 display the log-likelihood series of three types of burst-duration and sleep-duration obtained for AP overload scenario applying HMM and HMM-UBM methodologies. As it is shown in these figures, during the burst period the log-likelihood values drop drastically and in the sleep period it raises again to the normal level. The longer the burst period the wider is the valley shape in the log-likelihood series, and both HMM and HMM-UBM effectively detect heavy utilization periods in all these cases.

Figure 5.15 displays the boxplot diagram of the precision and recall results of RawData, PCA, HMM and HMM-UBM models. The low precision ratios of RawData and PCA show that this type of anomaly is not that straightforward to be detected directly from the data and needs more advanced techniques. The HMM



(A) burst-duration < sleep-duration (B) burst-duration = sleep-duration (C) burst-duration > sleep-duration

FIGURE 5.14: The log-likelihood series and detected anomalies of AP overload scenario (HMM-UBM).

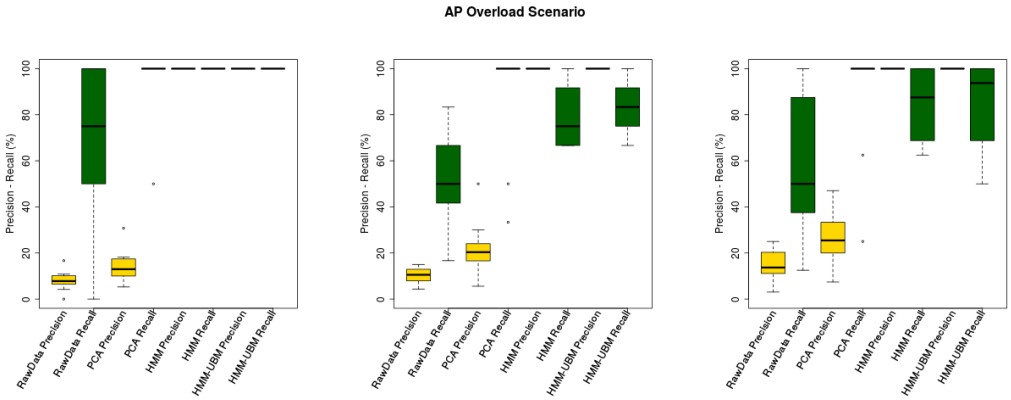


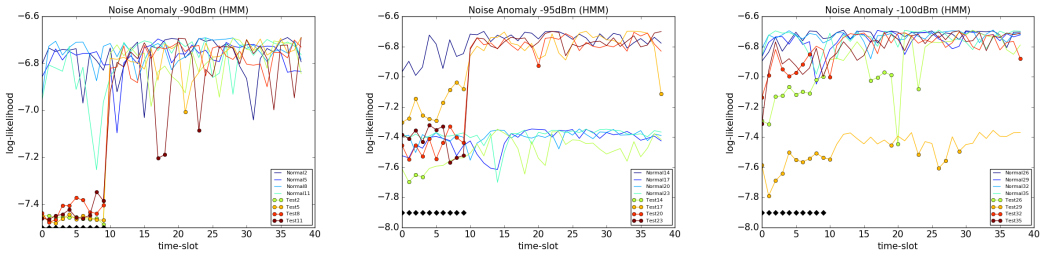
FIGURE 5.15: Precision and recall boxplot of RawData, PCA and HMM belong to AP overload scenario. Left: burstduration < sleepduration, middle: burstduration = sleepduration, right: burstduration > sleepduration.

and HMM-UBM results, both in precision and recall, outperform the baseline approaches.

Noise

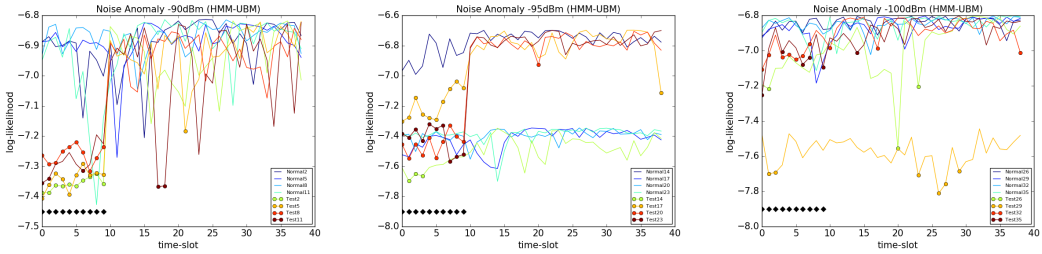
Thermal noise, cosmic background noise, and other random fluctuations of the electromagnetic field affect the quality of the communication channel. In the current experiment we change the level of noise power by adjusting the value of *IsotropicBackgroundNoise* parameter in the simulator. The default value of this parameter is set to -110dBm which is the minimum noise level in Wi-Fi networks 802.11 variants. We gradually increase the noise power to -90dBm and record the simulation results repeated 10 times for each experiment.

Figure 5.16 and 5.17 demonstrate the log-likelihood series of this anomalous scenario, and like previous cases the valley shapes represent the anomalies. The simulated anomalous period is during the first 10 time-slots which is marked with black diamond points. In the first experiment all the anomalous points are detected and the ratio of false positive is quite low. In the next two experiments the detection precision and sensitivity decline. The reason behind this downturn is that as the noise power decreases (higher negative value), it gets more difficult to detect the anomalous periods because the data becomes closer to the normal case (noise power of -110dBm).



(A) -90 dBm (B) -95 dBm (C) -100 dBm

FIGURE 5.16: The log-likelihood series and detected anomalies of noise scenario (HMM).



(A) -90 dBm (B) -95 dBm (C) -100 dBm

FIGURE 5.17: The log-likelihood series and detected anomalies of noise scenario (HMM-UBM).

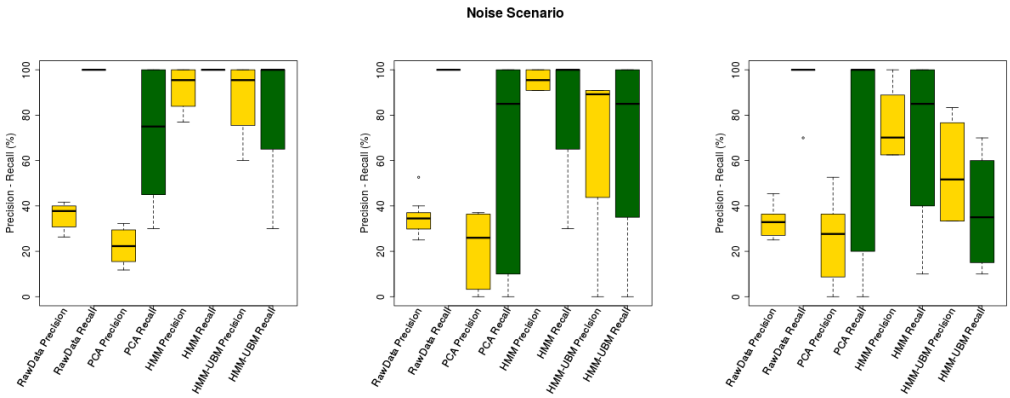
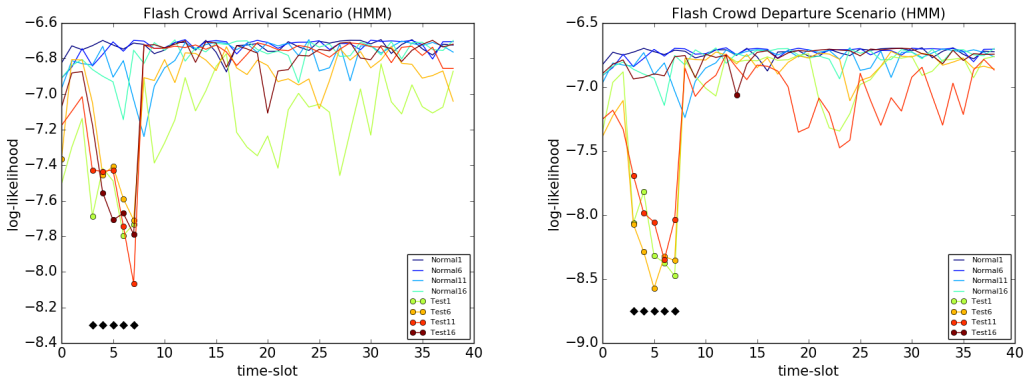


FIGURE 5.18: Precision and recall boxplot of RawData, PCA, HMM and HMM-UBM belong to noise scenario. Left: -90dBm, middle: -95dBm, right: -100dBm.

As the noise power increases, the packets are less likely to be received at the STAs. Therefore two data features are affected directly by the alteration of noise level: *OutputOctets* and *OutputPackets*. Hence the RawData detector is expected to produce satisfactory detection results. However, as Figure 5.18 shows, HMM and HMM-UBM models in all the experiments present higher precision values rather than RawData and PCA.

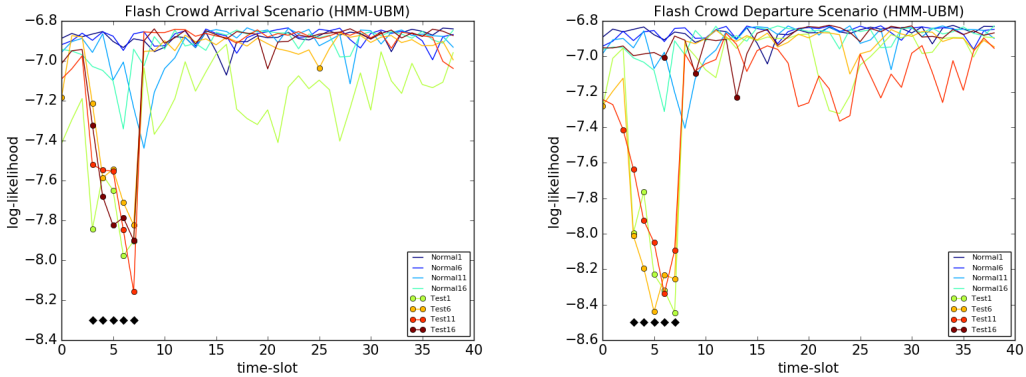
Flash Crowd

In wireless networks an unexpected surge of traffic occurs mostly due to the beginning or ending of an event when the majority of the wireless users abruptly enter



(A) Flash crowd arrival scenario (B) Flash crowd departure scenario

FIGURE 5.19: The log-likelihood series and detected anomalies of flash crowd scenario (HMM).



(A) Flash crowd arrival scenario (B) Flash crowd departure scenario

FIGURE 5.20: The log-likelihood series and detected anomalies of flash crowd scenario (HMM-UBM).

or leave a place and consequently associate to or disassociate from an AP. Such incidents are not necessarily an anomaly in terms of performance or connectivity issues, but could be considered as a sudden change to a routine network. To see whether the HMM and HMM-UBM models are able to detect such alterations in the normal usage pattern, we simulate this example in two experiments:

- Arrival: simultaneous association of 7 new nodes to the current AP.
- Departure: simultaneous disassociation of 7 existing nodes from the current AP.

Figure 5.19 and 5.20 represent the log-likelihood series of this scenario, detected anomalous points as colored circles, and simulated anomalies as black diamonds. Only in one test case in departure scenario which is related to *Rician Fading* path loss, anomalous period is not detected neither in HMM nor in HMM-UBM. In the rest of the experiments the anomaly detection technique performs accurately both in arrival and departure scenarios.

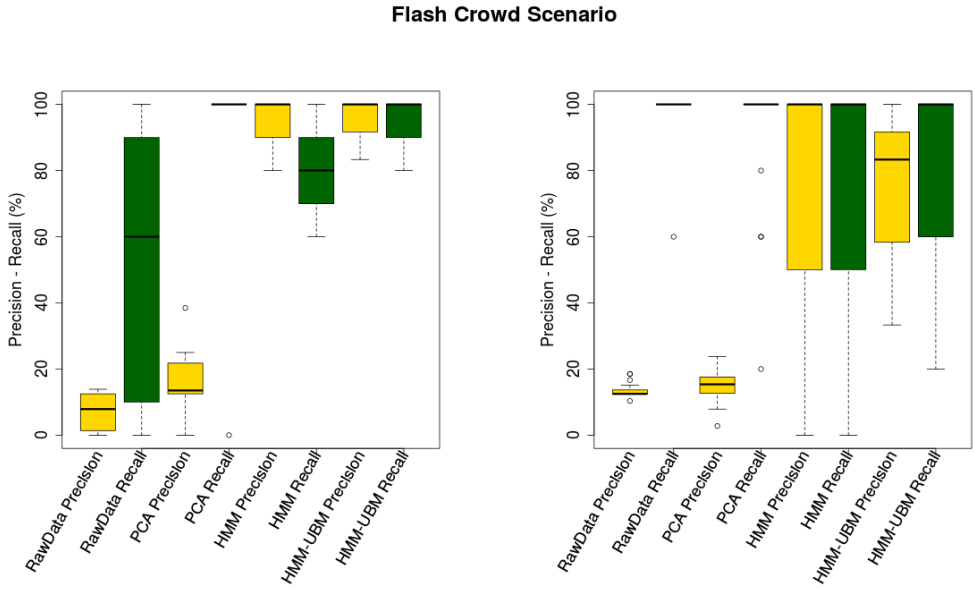


FIGURE 5.21: Precision and recall boxplot of RawData, PCA and HMM belong to flash crowd scenario. Left: arrival scenario, right: departure scenario.

As it is illustrated in the boxplot diagram of Figure 5.21, HMM and HMM-UBM easily outperform the RawData and PCA results in both *Arrival* and *Departure* scenarios. However, due to the aforementioned exception in the departure scenario, the arrival experiments achieve higher precision and recall.

5.3.4 Summary

The key contributions of this section consist of: 1) HMM modeling and threshold detection technique for anomaly detection, 2) proposing HMM-UBM technique for a robust initialization of the hidden states and unsupervised learning, 3) simulation of a small WLAN and a number of anomalous scenarios to evaluate the anomaly detection results.

The precision and recall outcomes of the anomalous cases are computed and compared to the baseline approaches (RawData and PCA). The experimental results show that HMM and HMM-UBM models are both capable of detecting a great portion of anomalies while producing only a trivial false positive ratio. This is promising for in HMM-UBM model all the data, regardless of being normal or containing anomalous events, is utilized to initialize the HMM model. Thus, in unsupervised learning, when the normal data is not known beforehand, HMM-UBM yields a robust model, as reliable as HMM initialized with normal data, for anomaly detection purposes.

5.4 Conclusion

In the first section of this chapter we proposed time-invariant (GMM) and time-variant (HMM) modeling approaches and utilized those models for anomaly detection evaluated on a RADIUS Testbed deployment with simulated anomalies.

We conducted an experiment as a case study on large data set of FEUP (3.2), and examined the goodness of fit using log-likelihood values. Results showed that

HMMs are more likely to provide a robust model to distinguish between their own pattern and an unfamiliar pattern rather than GMMs. Furthermore, we described the anomaly detection techniques by GMM and HMM, and we explored the addressed methodologies to detect anomalies at the same experimental data of the previous case study. We justified the detected anomalous points, however in absence of the ground truth in the large data set it was not possible to thoroughly evaluate the anomalous points, so we left the evaluation process for the Testbed experimental results. We measured the false positive rate (FPR), true negative rate (TNR), true positive rate (TPR), accuracy (ACC) and F1 score in normal and anomalous test data belong to Testbed data set (3.3). The experimental results demonstrated that HMM outperformed GMM in obtaining higher detection ratio while producing minor false alarm.

In the second section of this chapter we carried out HMM modeling and threshold detection technique for anomaly detection, in addition to proposing HMM-UBM technique for a robust initialization of the HMM models. The experimental results are performed on wireless simulation data set (3.4), and the precision and recall outcomes of the anomalous cases are computed and compared to the baseline approaches (RawData and PCA). The experimental results show that HMM and HMM-UBM models are both capable of detecting a considerable portion of anomalies while producing only a small false positive ratio. This is promising for in HMM-UBM model all the data, regardless of being normal or containing anomalous events, is utilized to initialize the HMM model. Thus, in unsupervised learning, when the normal data is not known beforehand, HMM-UBM yields a robust model, as reliable as HMM initialized with normal data, for anomaly detection purposes.

Chapter 6

Hidden Markov Models on Self-Organizing Maps for Spatio-Temporal Anomaly Detection

6.1 Introduction

This chapter introduces a hybrid integration of the Self-Organizing Map (SOM) and the Hidden Markov Model (HMM) for the purpose of anomaly detection in 802.11 wireless networks. The Self-Organizing Hidden Markov Model Map (SOHMMM) deals with the spatial connection of HMMs along with the inherent temporal dependency of data sequences. In essence, with each neuron of the SOHMMM lattice, an HMM is associated. In this thesis the SOHMMM algorithm is employed for anomaly detection in 802.11 wireless AP usage data. Furthermore we extend the online gradient descent unsupervised learning algorithm of SOHMMM for multivariate Gaussian emissions. Experimental analysis consist of two parts: *Synthetic Data* to investigate the accuracy and convergence of the SOHMMM algorithm, and *Wireless Simulation Data* to verify the significance and efficiency of the algorithm in anomaly detection. The sensitivity and specificity of the SOHMMM algorithm in anomaly detection is compared to two other approaches, namely HMM-UBM (HMM initialized with Universal Background Model addressed in chapter 5 section 5.3) and Z-SOHMMM (SOHMMM with zero neighborhood). The results from the wireless simulation experiments show that SOHMMM outperformed the aforementioned approaches in all the presented anomalous scenarios.

6.2 Background

In the previous chapter, we analyzed the AP usage data of 802.11 WLAN and proposed an anomaly detection technique for AP level anomalous events. We modeled the time-varying data sequences using Hidden Markov Models, and showed that HMMs are capable of detecting a considerable portion of anomalies while producing only a small false positive ratio. However, in these models the HMMs are learned individually (one per AP) and no connection is considered between them. In chapter 4, we studied individual HMMs versus single HMM (one model for all APs) and mixture of HMMs (groups of HMMs). While modeling an independent HMM per AP misses the opportunity to explore similarities between APs to improve learning, a single HMM for all APs loses the flexibility to learn AP specific behavior.

Using a single HMM common to all APs, which fail to have the flexibility to adapt to specificities of each AP, or using an HMM per AP learned independently from the others, failing to leverage the relations between observations of neighbouring APs, do not perform adequately. The UBM-HMM [11] is an improvement in the right direction but the relations between APs are only used in the initial phase, where one learns the UBM to then initialize the individual HMM models per AP. Thereafter the individual HMMs evolve independently, benefiting of the AP's own data only. Although this is a very sensible approach in the biometrics and speech modelling fields, where UBM-HMM has been used to robustly learn user specific models, in our setting it fails to properly explore the dependencies between data from APs with similar behavior.

In the current work we focus on actual proximity of APs as a determinant factor in connectivity and performance problems. Anomalous cases such as across AP vicinity interference, AP overloads or AP Shutdown/Halt could eventually affect the usage behavior of the other APs in the neighborhood. To take into account such behavioral changes in the local area and their respective influences, we employ the synergic approach of Self-Organizing Hidden Markov Model Map (SOHMMM) to exploit the semantic connectivity between adjacent HMMs.

The Self-Organizing Map is an artificial neural network that defines a nonlinear transformation from the input space to the set of nodes in the output space [81]. Each node or neuron in SOM is associated with a model of the input space. Through an unsupervised learning process, the models become tuned and organized in a lattice topology according to input patterns. In SOHMMM each neuron is literally associated with an HMM.

The training process of SOM and HMM subunits are in most cases disjoint and conducted independently. There are two main approaches regarding these hybrid techniques. First approach consider SOM as a front-end processor (e.g. vector quantization, preprocessing, feature extraction), and HMMs are then used in higher processing stages [81, 82]. While the second approach places the SOM on top of the HMM [80, 83].

In SOHMMM, the SOM unsupervised learning approach is well combined with the HMM dynamic programming technique. The structure of both corresponding components are unified in an integrated super-model. The presented online gradient descent unsupervised learning algorithm is inspired from the SOHMMM algorithm previously proposed in [80] and originated from [129]. We extend the model to fit the requirements of our anomaly detection problem and improve the algorithm in [80] for multivariate Gaussian emissions[1].

Thus the key contributions of the current work are: 1) extension of the previously proposed SOHMMM algorithm for multivariate Gaussian emissions, and 2) implementation, application, and validation of SOHMMM methodology on AP usage data for the purpose of anomaly detection in 802.11 wireless networks.

6.3 Self-Organizing Hidden Markov Model Map- Background and Notation

6.3.1 Estimating Model Parameters

For a more comprehensive definition of Hidden Markov Models please refer to Chapter 4 Section 4.3.1.

[1] [80] only addresses the discrete observations.

To adjust the model parameters (A, B, π) from the corpus of data O, an iterative procedure for optimization must be used, such as Expectation Maximization (EM) in the form of the Baum-Welch algorithm, or gradient descent technique. The Baum-Welch learning equations can be found in detail in [105, 106]. However, in certain cases the Baum-Welch algorithm can become problematic. For example when there is only a single training sequence, the algorithm resets a transition or emission probability to its expected frequency at each iteration. In such situations the Baum–Welch algorithm can cause abrupt jumps in parameter space, thus the procedure is not suitable for online learning e.g. for application after each training example [80, 129]. Another drawback of the Baum–Welch algorithm is the absorbing zero probabilities, meaning that once a transition or emission probability is set to 0, it remains equal to 0 in all iterations.

Consequently it is advantageous to resort to iterative and stochastic methods such as gradient descent. The gradient descent equations can be derived using a useful re-parameterization in the form of normalized exponentials [80, 129]:

$$a_{ij} = e^{w_{ij}} \Big/ \sum_{l=1}^{N} e^{w_{il}}, \qquad b_j(t) = e^{r_{jt}} \Big/ \sum_{l=1}^{M} e^{r_{jl}}, \qquad \pi_j = e^{u_j} \Big/ \sum_{l=1}^{N} e^{u_l} \tag{6.1}$$

The new variables can be arranged in a set of matrices, namely $W = \{w_{ij}\}$, $R = \{r_{jt}\}$ and $U = \{u_j\}$. In the online learning algorithm of SOHMMM these new variables get updated first and then the primary HMM model parameters are adapted subsequently.

6.3.2 Self-Organizing Hidden Markov Model Map

Studies show that the learning procedure of self-organization can be simplified into two partial processes [130, 131, 132]. For each input sample:

1. Find the best match neuron or the winner on the map by using the chosen similarity measure.
2. Update the model of the winner neuron as well as the neighborhood of neurons centered around the winner.

These two steps are repeated during the training process until the maximum number of iterations is reached or there is not enough improvement in the loss function. The model updating in the second step can be done incrementally after each input sample or in a batch process [132].

Figure 6.1 shows the lattice of SOHMMM, which is a mapping from the input observation sequence space onto a two-dimensional array of neurons. In general there are E neurons and with every neuron an HMM (λ_e) is associated. The shaded area is the topological neighborhood of the winner neuron which can be defined to any shape.

The unsupervised learning algorithm of SOHMMM is based on the optimization via stochastic gradient descent of an energy based cost function. The definition of the energy function given in [80] is:

$$\xi = \left\langle \sum_{d=1}^{E} K(O, \lambda_d) \sum_{e=1}^{E} \{-h_{de} P(O|\lambda_e)\} \right\rangle \tag{6.2}$$

where $\langle \cdots \rangle$ denotes averaging over the distribution of input sequences, h_{de} corresponds to the neighborhood function of the lattice, and $P(O|\lambda_e)$ is likelihood or the probability of O given λ_e,

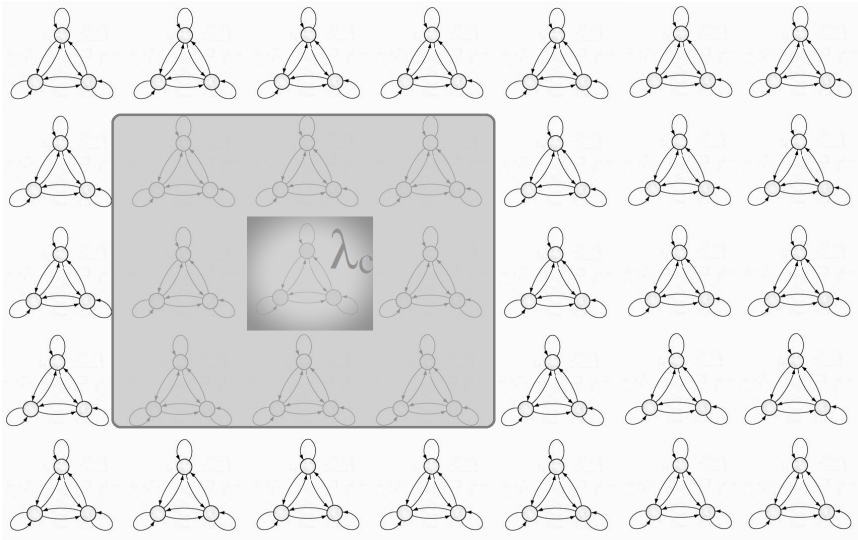


FIGURE 6.1: SOHMMM lattice. Each neuron is associated with a 3-state HMM. The HMM model in highlight is supposed to be the winner HMM (λ_c) and the shaded area is the neighborhood of the winner HMM. [80]

and

$$K(O,\lambda_d) = \begin{cases} 1, & c = \underset{d}{\operatorname{argmax}} \sum_{e=1}^{E} h_{de}(y) P(O|\lambda_e)|_y. \\ 0, & \text{otherwise.} \end{cases} \tag{6.3}$$

The energy function in 6.2 measures the adaptability and capability of SOHMMM to describe a corpus of observation sequences and is continuous for finite sets of sequences.

Given an observation sequence O, the aforementioned energy function yields the sample function:

$$\xi(O) = \sum_{d=1}^{E} K(O,\lambda_d) \sum_{e=1}^{E} \{-h_{de} P(O|\lambda_e)\}. \tag{6.4}$$

Each online learning step conforms to a local gradient descent on such a sample. A stochastic gradient descent with respect to the parameter x of the SOHMMM has the following form:

$$x^{(next)} = x^{(now)} - \eta \partial \xi(O)/\partial x|_{x=x^{(now)}}. \tag{6.5}$$

where η is the learning rate, and can be fixed or adjusted during the learning process. Having performed the differentiation we have:

$$\begin{aligned} \partial \xi(O)/\partial x = & \sum_{d=1}^{E} \partial K(O,\lambda_d)/\partial x \sum_{e=1}^{E} \{-h_{de} P(O|\lambda_e)\} \\ & + \sum_{d=1}^{E} K(O,\lambda_d) \sum_{e=1}^{E} \{-h_{de} \partial P(O|\lambda_e)/\partial x\}. \end{aligned} \tag{6.6}$$

The first term on the right hand side vanishes according to [133] for a detailed evaluation/justification, and the second term on the right hand side yields the desired learning rule:

$$x^{(next)} = x^{(now)} + \eta \sum_{d=1}^{E} K(O,\lambda_d) \sum_{e=1}^{E} \{h_{de} \partial P(O|\lambda_e)/\partial x|_{x=x^{(now)}}. \tag{6.7}$$

Having used stochastic gradient descent on an energy function optimizes a global measure of performance throughout the learning process.

Adjusting the parameters of the HMM in order to minimize the corresponding energy function is what SOHMMM algorithm accomplishes during an iterative procedure. However, as already stated in section 6.3.1, the parameters of the HMM are adjusted through the parameters of interest, namely $W^{(e)}$, $R^{(e)}$ and $U^{(e)}$. The best matching HMM (winner) corresponds to the maximum weighted likelihood and is given by:

$$c = \underset{d}{\operatorname{argmax}} \sum_{e=1}^{E} h_{de}(y) P(O|\lambda_e)|_y \tag{6.8}$$

where $y = 1, 2, 3, ...$ is the discrete time coordinate. With the use of 6.7 and Proposition 1 (8.2) in Appendix, the online gradient descent equation with respect to w_{ij} is:

$$\begin{aligned} w_{ij}^{(e)}(y+1) &= w_{ij}^{(e)}(y) + \eta(y) h_{ce}(y) \\ &\cdot \left[a_{ij} \sum_{l=1}^{T-1} \left[\alpha_l(i) b_j(o_{l+1}) \beta_{l+1}(j) - \alpha_l(i) \beta_l(i) \right] \Big|_{\lambda_e, y} \right], \\ &\qquad 1 \leq i \leq N,\ 1 \leq j \leq N \end{aligned} \tag{6.9}$$

Similarly by using 6.7 and Proposition 2 (8.4) in Appendix, the stochastic learning rule with respect to r_{jt} is:

$$\begin{aligned} r_{jt}^{(e)}(y+1) &= r_{jt}^{(e)}(y) + \eta(y) h_{ce}(y) \\ &\cdot \left[\sum_{l=1}^{T} \left[I\{o_l = t|\lambda\} \alpha_l(j) \beta_l(j) - b_j(t) \alpha_l(j) \beta_l(j) \right] \Big|_{\lambda_e, y} \right], \\ &\qquad 1 \leq j \leq N,\ 1 \leq t \leq M \end{aligned} \tag{6.10}$$

Eventually, with the use of 6.7 and Proposition 3 (8.6) in Appendix, the stochastic gradient descent equation with respect to u_j is:

$$\begin{aligned} u_j^{(e)}(y+1) &= u_j^{(e)}(y) + \eta(y) h_{ce}(y) \\ &\cdot \left[\left[\pi_j b_j(o_1) \beta_1(j) - \pi_j P(O|\lambda) \right] \Big|_{\lambda_e, y} \right], \\ &\qquad 1 \leq j \leq N \end{aligned} \tag{6.11}$$

The function $\eta(y)$ denotes a scalar learning rate factor ($0 < \eta(y) < 1$), and can be fixed or decreasing monotonically as time passes. The function $h_{ce}(y)$ is the neighborhood function, a smoothing kernel defined over the lattice points and it is required that $h_{ce}(y) \to 0$ when $y \to \infty$. The smoother neighborhood kernel defined in [80] is in terms of the Gaussian function:

$$h_{ce}(y) = exp(-||\delta_c - \delta_e||^2 / 2\sigma^2(y))) \tag{6.12}$$

where $\delta_c, \delta_e \in \mathbb{R}^2$ are the location vectors of HMM λ_c and λ_e on the array, and $\sigma(y)$ corresponds to the radius of the neighborhood.

6.3.3 The SOHMMM Learning Algorithm

We denote the set of D observation sequences as $OS = O^1, O^2, \cdots, O^D$, where $O^{(d)} = \{o_1, o_2, \cdots, o_{T_d}\}$ is the d-th observation sequence. In discrete mode each observation is supposed to be a value from the set of alphabets $V = \{v_1, v_2, \cdots, v_M\}$, and T_d is the number of observations in the sequence $O^{(d)}$.

The SOHMMM consists of E HMMs that are assigned to the nodes of a two-dimensional lattice. There are E_1 HMMs per lattice row and E_2 HMMs per lattice column ($E_1 \cdot E_2 = E$). The SOHMMM online gradient descent unsupervised learning algorithm in discrete mode, as presented in [80], is demonstrated in Algorithm 1.

6.4 Extension of the SOHMMM Algorithm

The SOHMMM algorithm addressed in [80] only deals with the discrete observation setting. In this section we extend the aforementioned learning algorithm to incorporate the multivariate Gaussian emissions, as well. Moreover, we adapt this new algorithm to conform to the requirements of our anomaly detection problem in AP usage data.

6.4.1 SOHMMM Algorithm for Gaussian Observations

Univariate observations

Based on Lemma 2 (8.3) in Appendix 8 we have:

$$\frac{\partial P(O|\lambda)}{\partial b_x(y)} = \sum_{l=1}^{\mathsf{T}} \frac{1}{b_x(o_l)} I\{o_l = y|\lambda\}\alpha_l(x)\beta_l(x), \tag{6.13}$$

where $b_x(y)$ is the emission probability of observation y, which is 1-dimensional, given the state x. In case of continuous Gaussian emissions, we have $y|x \sim \mathcal{N}(y|\mu_x, \sigma_x^2)$, thus:

$$b_x(y) = P(y|x) = \mathcal{N}(y|\mu_x, \sigma_x^2) = \frac{1}{\sqrt{2\pi}\sigma_x} \exp\left(-\frac{(y-\mu_x)^2}{2\sigma_x^2}\right), \tag{6.14}$$

$$\frac{\partial b_x(y)}{\partial \mu_x} = \frac{y-\mu_x}{\sigma_x^2}\mathcal{N}(y|\mu_x, \sigma_x^2), \tag{6.15}$$

$$\frac{\partial b_x(y)}{\partial \sigma_x} = \left(\frac{(y-\mu_x)^2}{\sigma_x^3} - \frac{1}{\sigma_x}\right)\mathcal{N}(y|\mu_x, \sigma_x^2). \tag{6.16}$$

We also need the derivative of $P(O|\lambda)$ with respect to the parameters μ_x and σ_x, which may be obtained using the chain rule:

$$\frac{\partial P(O|\lambda)}{\partial \theta} = \sum_{l=1}^{T} \frac{\partial P(O|\lambda)}{\partial b_x(o_l)} \frac{\partial b_x(o_l)}{\partial \theta} = \sum_{l=1}^{T} \frac{1}{b_x(o_l)} \frac{\partial b_x(o_l)}{\partial \theta} \alpha_l(x)\beta_l(x), \tag{6.17}$$

where θ can be either μ_x or σ_x. Now, using these equations together with Equation 6.7, we have all that we need to update parameters μ_x and σ_x.

Multivariate observations

In this setting, observations $\boldsymbol{y}$ are d-dimensional, so we need to use a multivariate Gaussian to model the distribution of $\boldsymbol{y}|x$:

$$b_x(\vec{y}) = P(\vec{y}|x) = \mathcal{N}(\vec{y}|\vec{\mu}_x, \Sigma_x) = \frac{1}{\sqrt{(2\pi)^n|\Sigma_x|}} \exp\left(-\frac{1}{2}(\vec{y}-\vec{\mu}_x)^{\mathsf{T}}\Sigma_x^{-1}(\vec{y}-\vec{\mu}_x)\right), \tag{6.18}$$

where $\vec{\mu}_x \in \mathbb{R}^d$ and $\Sigma_x \in \mathbb{R}^{d\times d}$ and is symmetric and positive semidefinite.

The fact that Σ_x is positive semidefinite poses some difficulties to gradient-based optimization, so we use the fact that any matrix Σ_x is positive semidefinite if and only if it can be factorized as:

$$\Sigma_x = S_x^{\mathsf{T}} \cdot S_x, \tag{6.19}$$

Algorithm 1 SOHMMM Learning Algorithm for Discrete HMMs [80]

1: **for** y=1 to trainIterations **do**

2: $O \leftarrow selectObservationSequence(HMM\_list, Observation\_list)$

3: **for** e=1 to E **do**

4: $\alpha_1^{(e)}(j) = \pi_j^{(e)} b_j^{(e)}(o_1), \quad 1 \leq j \leq N;$

5: $\alpha_{t+1}^{(e)}(j) = \left[\sum_{i=1}^{N} \alpha_t(e)^{(i)} a_{ij}^{(e)}\right] b_j^{(e)}(o_{t+1}), \quad 1 \leq t \leq T-1, \quad 1 \leq j \leq N;$

6: $\beta_T^{(e)}(j) = 1, \quad 1 \leq j \leq N;$

7: $\beta_t^{(e)}(j) = \sum_{i=1}^{N} a_{ji}^{(e)} b_i^{(e)}(o_{t+1}) \beta_{t+1}^{(e)}(i), \quad T-1 \leq t \leq 1, \quad 1 \leq j \leq N;$

8: $c \leftarrow \underset{e}{\operatorname{argmax}} \sum_{\varepsilon=1}^{E} exp(-||\delta_e - \delta_\varepsilon||^2/2\sigma^2(y)))P(O|\lambda_\varepsilon);$

9: **for** e=1 to E **do**

10: $w_{ij}^{(e)} \leftarrow w_{ij}^{(e)} + \eta(y) exp(-||\delta_e - \delta_\varepsilon||^2/2\sigma^2(y))$

11: $\cdot \left[a_{ij} \sum_{l=1}^{T-1} \left[\alpha_l(i) b_j(o_{l+1}) \beta_{l+1}(j) - \alpha_l(i)\beta_l(i)\right] \Big|_{\lambda_e} \right], \quad 1 \leq i \leq N, 1 \leq j \leq N;$

12: $r_{jt}^{(e)} \leftarrow r_{jt}^{(e)} + \eta(y) exp(-||\delta_e - \delta_\varepsilon||^2/2\sigma^2(y))$

13: $\cdot \left[\sum_{l=1}^{T} \left[I\{o_l = t|\lambda\}\alpha_l(j)\beta_l(j) - b_j(t)\alpha_l(j)\beta_l(j)\right] \Big|_{\lambda_e} \right], 1 \leq j \leq N, 1 \leq t \leq M;$

14: $u_j^{(e)} \leftarrow u_j^{(e)} + \eta(y) exp(-||\delta_e - \delta_\varepsilon||^2/2\sigma^2(y))$

15: $\cdot \left[\left[\pi_j b_j(o_1)\beta_1(j) - \pi_j P(O|\lambda)\right] \Big|_{\lambda_e} \right], \quad 1 \leq j \leq N;$

16: $a_{ij}^{(e)} \leftarrow e^{w_{ij}^{(e)}} \Big/ \sum_{l=1}^{N} e^{w_{il}^{(e)}}, \quad 1 \leq i \leq N;, 1 \leq j \leq N$

17: $b_j^{(e)}(t) \leftarrow e^{r_{jt}^{(e)}} \Big/ \sum_{l=1}^{M} e^{r_{jl}^{(e)}}, \quad 1 \leq j \leq N;, 1 \leq t \leq M$

18: $\pi_j^{(e)} \leftarrow e^{u_j^{e)}} \Big/ \sum_{l=1}^{N} e^{u_{l(e)}}, \quad 1 \leq j \leq N;$

where $S_x \in \mathbb{R}^{n \times n}$. Using this decomposition, equation 6.18 becomes:

$$b_x(\vec{y}) = \frac{1}{\sqrt{(2\pi)^n |S_x|^2}} \exp\left(-\frac{1}{2}(\vec{y} - \vec{\mu}_x)^\intercal (S_x^\intercal S_x)^{-1} (\vec{y} - \vec{\mu}_x)\right). \tag{6.20}$$

Now, we can compute our desired derivatives:

$$\frac{\partial b_x(\vec{y})}{\vec{\mu}_x} = \mathcal{N}(\vec{y}|\vec{\mu}_x, S_x^\intercal S_x)(S_x^\intercal S_x)^{-1}(\vec{y} - \vec{\mu}_x), \tag{6.21}$$

$$\frac{\partial b_x(\vec{y})}{S_x} = \mathcal{N}(\vec{y}|\vec{\mu}_x, S_x^\intercal S_x)(-S_x^{-\intercal} + S_x^{-\intercal}(\vec{y} - \vec{\mu}_x)(\vec{y} - \vec{\mu}_x)^\intercal (S_x^\intercal S_x)^{-1}), \tag{6.22}$$

Thenceforth we apply Equations (6.17) and (6.7) to update the parameters $\vec{\mu}_x$ and S_x:

$$\begin{aligned}
\vec{\mu}_x^{(next)} &= \vec{\mu}_x^{(now)} + \eta(y) exp(-||\delta_e - \delta_\varepsilon||^2/2\sigma^2(y)) \\
&\cdot \left[\frac{1}{\mathcal{N}(\vec{y}|\vec{\mu}_x, S_x^\intercal S_x)} \mathcal{N}(\vec{y}|\vec{\mu}_x, S_x^\intercal S_x)(S_x^\intercal S_x)^{-1}(\vec{y} - \vec{\mu}_x)\alpha_l(x)\beta_l(x)\right] \\
&= \vec{\mu}_x^{(now)} + \eta(y) exp(-||\delta_e - \delta_\varepsilon||^2/2\sigma^2(y)) \cdot \left[(S_x^\intercal S_x)^{-1}(\vec{y} - \vec{\mu}_x)\alpha_l(x)\beta_l(x)\right]
\end{aligned} \tag{6.23}$$

$$\begin{aligned}
S_x^{(next)} &= S_x^{(now)} + \eta(y) exp(-||\delta_e - \delta_\varepsilon||^2/2\sigma^2(y)) \\
&\cdot \left[\frac{1}{\mathcal{N}(\vec{y}|\vec{\mu}_x, S_x^\intercal S_x)} \mathcal{N}(\vec{y}|\vec{\mu}_x, S_x^\intercal S_x)(-S_x^{-\intercal} + S_x^{-\intercal}(\vec{y} - \vec{\mu}_x)(\vec{y} - \vec{\mu}_x)^\intercal (S_x^\intercal S_x)^{-1})\alpha_l(x)\beta_l(x)\right] \\
&= S_x^{(now)} + \eta(y) exp(-||\delta_e - \delta_\varepsilon||^2/2\sigma^2(y)) \\
&\cdot \left[(-S_x^{-\intercal} + S_x^{-\intercal}(\vec{y} - \vec{\mu}_x)(\vec{y} - \vec{\mu}_x)^\intercal (S_x^\intercal S_x)^{-1})\alpha_l(x)\beta_l(x)\right]
\end{aligned} \tag{6.24}$$

Eventually in Algorithm 1 instead of updating r_{jt} probabilities of matrix R, we directly update μ_x and σ_x in univariate observations or $\vec{\mu}_x$ and Σ_x in multivariate observations.

Thus the SOHMMM online gradient descent unsupervised learning algorithm for multivariate Gaussian emissions is formulated in Algorithm 2.

6.4.2 Anomaly Detection with the Extended SOHMMM Algorithm

In SOHMMM adapted to the problem of anomaly detection in AP usage data, the learning process of self-organization (addressed in Section 6.3.2) is elaborated as follows. In this work the models associated with the SOM neurons are three-state Hidden Markov Models (according to our best practice in our previous works in [10, 11]). As the new observation sequence arrives, an HMM with the highest log-likelihood value is selected as the winner model. In the AP usage data, as the newly arrived sequence belongs to a pre-determined AP, in most cases the winner model is related to the same AP that has originated the observation sequence. However, it is not always the case and the competition defines the winner HMM eventually. Thereafter, the HMM model of the winner AP is updated by the new data sequence and the HMMs in the neighborhood of the winner AP get updated, as well.

Algorithm 2 SOHMMM Learning Algorithm for Continuous HMMs

1: **for** y=1 to trainIterations **do**

2: $O \leftarrow selectObservationSequence(HMM\_list, Observation\_list)$

3: **for** e=1 to E **do**

4: $\alpha_1^{(e)}(j) = \pi_j^{(e)} b_j^{(e)}(o_1), \quad 1 \leq j \leq N;$

5: $\alpha_{t+1}^{(e)}(j) = \left[\sum_{i=1}^{N} \alpha_t(e)^{(i)} a_{ij}^{(e)}\right] b_j^{(e)}(o_{t+1}), \quad 1 \leq t \leq T-1, \quad 1 \leq j \leq N;$

6: $\beta_T^{(e)}(j) = 1, \quad 1 \leq j \leq N;$

7: $\beta_t^{(e)}(j) = \sum_{i=1}^{N} a_{ji}^{(e)} b_i^{(e)}(o_{t+1}) \beta_{t+1}^{(e)}(i), \quad T-1 \leq t \leq 1, \quad 1 \leq j \leq N;$

8: $c \leftarrow \underset{e}{\operatorname{argmax}} \sum_{\varepsilon=1}^{E} exp(-||\delta_e - \delta_\varepsilon||^2/2\sigma^2(y)))P(O|\lambda_\varepsilon);$

9: **for** e=1 to E **do**

10: $w_{ij}^{(e)} \leftarrow w_{ij}^{(e)} + \eta(y) exp(-||\delta_e - \delta_\varepsilon||^2/2\sigma^2(y))$

11: $\cdot \left[a_{ij} \sum_{l=1}^{T-1} \left[\alpha_l(i) b_j(o_{l+1}) \beta_{l+1}(j) - \alpha_l(i)\beta_l(i)\right] \Big|_{\lambda_e} \right], \quad 1 \leq i \leq N, 1 \leq j \leq N;$

12: $\vec{\mu}_x^{(next)} = \vec{\mu}_x^{(now)} + \eta(y) exp(-||\delta_e - \delta_\varepsilon||^2/2\sigma^2(y))$

13: $\cdot \left[(S_x^\intercal S_x)^{-1} (\vec{y} - \vec{\mu}_x) \alpha_l(x) \beta_l(x) \right]$

14: $S_x^{(next)} = S_x^{(now)} + \eta(y) exp(-||\delta_e - \delta_\varepsilon||^2/2\sigma^2(y))$

15: $\cdot \left[(-S_x^{-\intercal} + S_x^{-\intercal} (\vec{y} - \vec{\mu}_x)(\vec{y} - \vec{\mu}_x)^\intercal (S_x^\intercal S_x)^{-1}) \alpha_l(x) \beta_l(x) \right]$

16: $\Sigma_x^{(next)} = S_x^{(next)^\intercal} \cdot S_x^{(next)}$

17: $u_j^{(e)} \leftarrow u_j^{(e)} + \eta(y) exp(-||\delta_e - \delta_\varepsilon||^2/2\sigma^2(y))$

18: $\cdot \left[\left[\pi_j b_j(o_1) \beta_1(j) - \pi_j P(O|\lambda)\right] \Big|_{\lambda_e} \right], \quad 1 \leq j \leq N;$

19: $a_{ij}^{(e)} \leftarrow e^{w_{ij}^{(e)}} \Big/ \sum_{l=1}^{N} e^{w_{il}^{(e)}}, \quad 1 \leq i \leq N;, 1 \leq j \leq N$

20: $\pi_j^{(e)} \leftarrow e^{u_j^{(e)}} \Big/ \sum_{l=1}^{N} e^{u_{l(e)}}, \quad 1 \leq j \leq N;$

At this point it should be noted that the APs in the vicinity of the winner AP get updated to some extent relative to their proximity (or similarity) to the winner AP. In our case, the neighborhood area has an irregular shape and contains the first-level adjacent APs in the vicinity of the winner AP. In the wireless simulation experiments in Section 6.5.2, as the location of the APs are already determined in the wireless ground, the Euclidean distances between all APs are calculated and kept in a complete graph. Then a filter is applied to update only APs with a certain distance from the winner AP (in this case half of the maximum distance the distance graph). During the learning process, the nearby HMMs up to a certain distance activate each other to gain some information from the new observation sequence. As the distant HMMs can only gain insignificant amount of information from the new observation sequence, we utilized the aforementioned filter to avoid updating very distant HMMs and speed up the process. The neighborhood function that we used in our algorithm is based on the relative distance of the winner AP and all its adjacent neighbors:

$$h_{ce}(y) = exp(-10 \cdot d(c,e) / \sum_{i=1}^{N} d(i,e)) \tag{6.25}$$

where $d(c,e)$ is the Euclidean distance between AP_c and the winner AP, AP_e, N is the number of adjacent APs of AP_e.

6.5 Experimental Study

In this section we consider two types of experiments to analyze the capabilities of SOHMMM algorithm in nonlinear projection and unsupervised clustering. We first validate the accuracy and convergence of SOHMMM using *synthetic data*, and then explore its significance and efficiency in anomaly detection using *wireless simulation data*.

6.5.1 Synthetic Data

In this experiment we generate observations from 2 reference HMMs, with one third of the observations coming from one of the models, and the remaining two thirds from the second reference model. Then we train a SOHMMM with 6 nodes, randomly initialized, with the data from the reference models. We expect that the SOHMMM nodes will converge to the reference models, with majority of nodes grouping around the dominant reference model.The SOHMMM nodes (HMMs initialized randomly) are organized in a 2×3 rectangular lattice, displayed in Figure 6.2. This figure shows the position and connections of the HMMs. The Euclidean distance between adjacent HMMs is set to 1 and the other distances get computed accordingly. For example distance between *hmm0* and *hmm4* is 2, and between *hmm0* and *hmm3* is $\sqrt{2}$.

Equation 6.26 demonstrates how the distance (dis-similarity) between two HMMs is estimated, in this example λ_1 and λ_2:

$$D(\lambda_1, \lambda_2) = \frac{1}{T}[logP(O_T|\lambda_1) - logP(O_T|\lambda_2)]. \tag{6.26}$$

where $O_T = o_1, o_2, ..., o_T$ is a sequence of observations generated by λ_1 and T is the number of observations in O_T. Equation 6.26 is a Monte Carlo approximation of the Kullback-Leibler divergence between two HMMs [134], and is a measure of how well model λ_1 matches observations generated by model λ_2, relative to how well

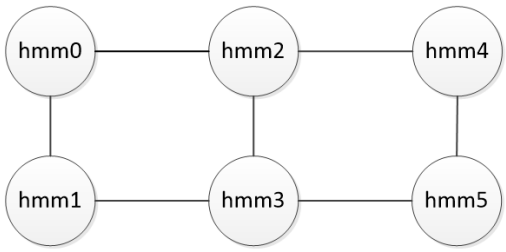


FIGURE 6.2: 2×3 rectangular lattice.

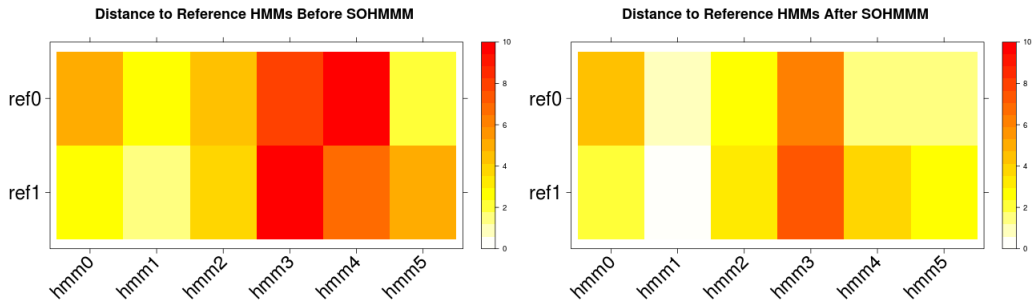


FIGURE 6.3: Monte Carlo approximation of the Kullback-Leibler divergence between random HMMs and reference HMMs before and after applying SOHMMM algorithm.

model λ_2 matches observations generated by itself [106]. The distance measure of Equation 6.26 is non-symmetric. Hence the symmetrized version of this measure is:

$$D_s(\lambda_1, \lambda_2) = \frac{D(\lambda_1, \lambda_2) + D(\lambda_2, \lambda_1)}{2}. \tag{6.27}$$

Figure 6.3 demonstrates the initial and final distances between the random HMMs and reference HMMs upon applying the SOHMMM algorithm. The training set contains 60 observation sequences from the two reference HMMs (*ref0* model generates 40 observation sequences and *ref1* generates 20). We train the random HMMs with these observation sequences.

As the heatmap plots of Figure 6.3 show, initially *hmm4* is assigned to *ref1* cluster as there is a shorter distance between these two models, and after applying the algorithm *hmm4* gets closer to *ref0*. As another example *hmm3* is closer to *ref0* before and after applying the algorithm, however the overall distance to the reference models decreases and *hmm3* gets closer to both of them maintaining its relative distance to the references. The minimum distance between a given random HMM and the reference HMMs define the cluster that HMM belongs to. In this experiment the random HMMs belong to $(ref1, ref1, ref1, ref0, ref1, ref0)$ and $(ref1, ref1, ref0, ref0, ref0, ref0)$ clusters before and after applying the algorithm, respectively. The later clusters

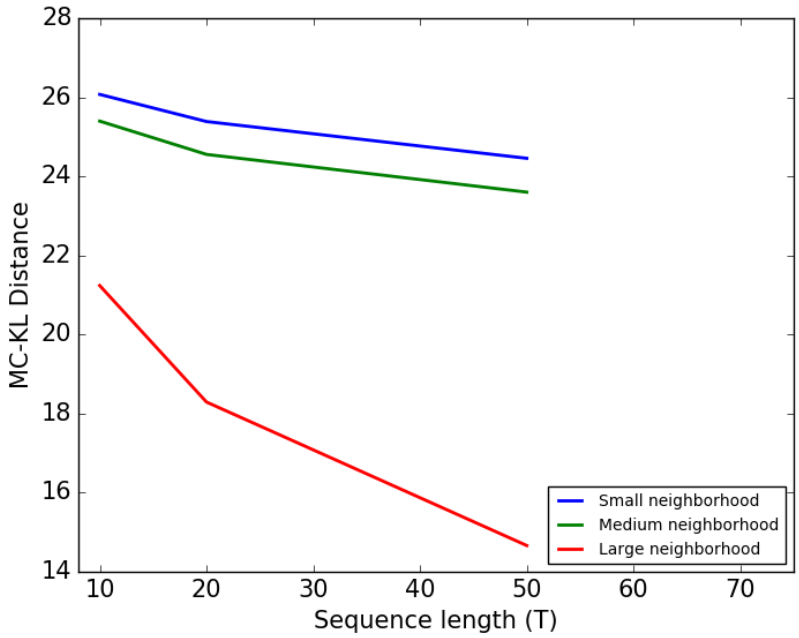


FIGURE 6.4: Monte Carlo approximation of the Kullback-Leibler divergence between random HMMs and reference HMMs with different neighborhood sizes and sequence lengths.

show that $\{hmm2, hmm3, hmm4, hmm5\}$ belong to $ref0$ cluster, the dominant reference models, and $\{hmm0, hmm1\}$ are assigned to $ref1$ cluster, the reference model generating less data. Having analyzed the Euclidean distances of HMM nodes in Figure 6.2 shows that nearby HMMs are grouped in the same cluster.

We repeat the experiment for various observation sequence lengths and different types of neighborhood in terms of vicinity. We select large neighborhood and update the winner neuron in addition to all the other neurons in the SOM lattice based on their relative distances addressed in Equation 6.25. The large neighborhood experiment employs the SOHMMM algorithm in its original form. In other experiments we make some changes to the SOHMMM algorithm in terms of nearby HMMs that need to be updated. In medium neighborhood experiment the winner HMM is updated in addition to its adjacent neighbors. In small neighborhood experiment only the winner HMM gets updated. The small neighborhood experiment is equivalent to Z-SOHMMM experiment.

Figure 6.4 displays the Monte Carlo approximation of the Kullback-Leibler divergence between random HMMs and reference HMMs with large, medium, and small neighborhood sizes and sequence lengths of 10, 20, and 50. In each experiment the sum of random HMMs' distance with the assigned reference HMM is computed. As Figure 6.4 shows, the sum of distances to the assigned reference HMMs decreases as the observation sequence length increases. Also the lower distance values are obtained in large neighborhood experiment rather than medium and small neighborhood experiments. It shows that the SOHMMM algorithm provides better estimation of the reference models including all the HMMs in the vicinity. The heatmap plots of Figure 6.3 belong to the large neighborhood experiment with observation sequence length of 50 that produces the best overall distance estimate in Figure 6.4.

The learning rate of the SOHMMM algorithm follows a decay function which

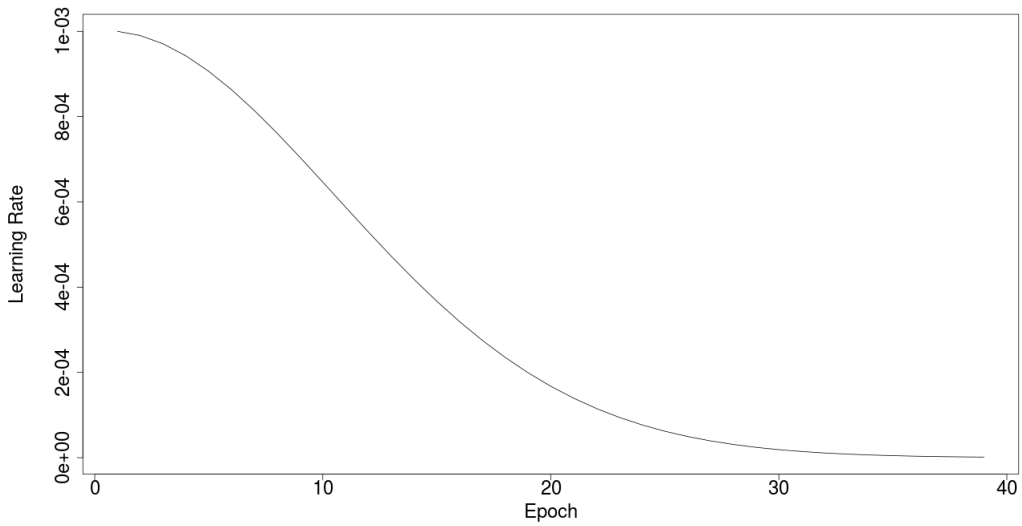


FIGURE 6.5: Learning rate decay over time.

is computed according to Equation 6.28. The intuition behind slowly reducing the learning rate is to speed up the training and avoid large steps as the learning process approaches the end. Thus, the convergence becomes more feasible.

$$learning\_rate = learning\_rate \times (1./(1. + decay\_rate \times epoch\_number)) \tag{6.28}$$

The initial value of the learning rate for this experiment is set to 1e−2 and the decay rate is tuned to 1e−2. The learning rate evolution over time is depicted in Figure 6.5.

To inspect the convergence of the algorithm, the evolution of the SOHMMM's loss function, previously calculated in Equation 6.2, is demonstrated in Figure 6.6. The SOHMMM evolves rapidly, yet as it reaches the approximate shape of the data, the rate of changes decreases. We performed the algorithm for 40 epochs, however the loss value declines drastically before the 20th epoch, so performing the training phase for about 20 epochs produces promising results. In order to speed up the training process we considered 10 epochs for each experiment.

6.5.2 Wireless Simulation Data

Our simulation consists of one normal scenario and four anomalous scenarios: AP Shutdown/Halt, AP Overload, Noise, and Flash Crowd. In a normal scenario, there are 10 access points and 100 wireless stations (STA). Each STA is initially associated to one of the available APs depending on its location. During the simulation STAs are handed over to other APs, based on their mobility models, when moving around the simulation ground. Furthermore, according to the defined traffic plans, each node sends and receives packets to the existing servers. More details on the mobility models of the wireless stations, traffic generation, available servers, and path loss models can be found in Chapter 3 Section 3.4.

In our HMM approach one observation sequence contains 40 consecutive time-slots of 15s simulation time each. We simulated 15 instances of 3000s simulation time for normal scenario, and 5 instances of 3000s simulation time for each of the anomalous scenarios.

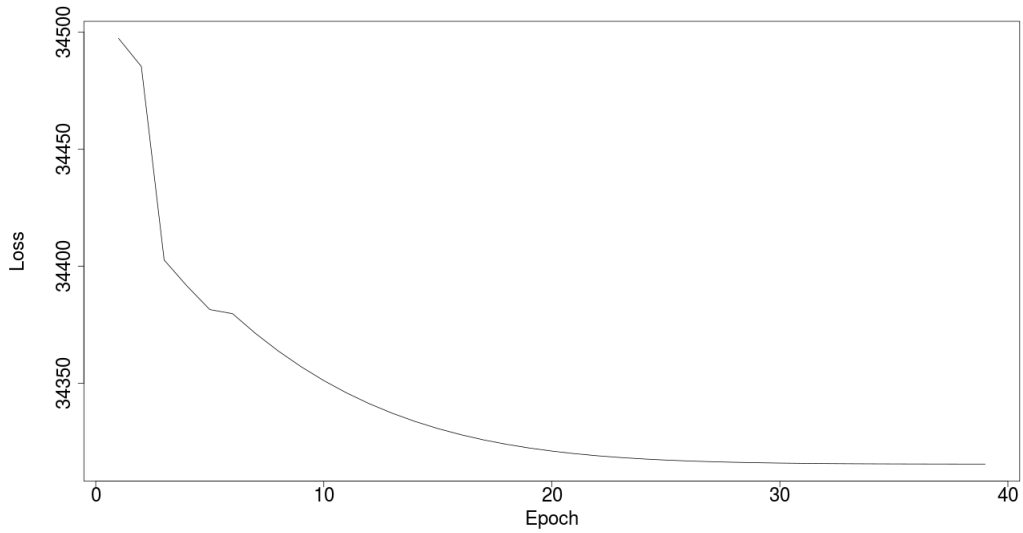


FIGURE 6.6: Convergence of the SOHMMM's loss.

Our data set is divided into train set (80%) and test set (20%). We compare the SOHMMM algorithm to two other approaches: 1) Hidden Markov Model initialized with Universal Background Model (HMM-UBM), addressed in details in chapter 5 section 5.3, and 2) SOHMMM algorithm with zero neighborhood which is just utilizing the incremental learning part of the SOHMMM algorithm without updating the hmms in the vicinity (Z-SOHMMM). To initialize the HMM with UBM, the amount of normal and anomalous data need to be equivalent. So we use 5 weeks of simulation data from the normal data set, and 1 week of each anomalous cases (5X), which in overall provides us with 10 weeks data, and each week contains 5 working days. In anomalous cases the time and period of anomalies are different in each day. We then use one more week of each scenario to train the model and use one remaining week of each case to test the model. The results are represented as Receiver Operating Characteristic (ROC) curves [135], and belong to the test set only. The learning rate follows a decay function defined in Equation 6.28, and its initial value is set to 1e−3. The duration of the training phase is set to 10 epochs for SOHMMM and Z-SOHMMM algorithms.

AP Shutdown/Halt

In our simulation, we reproduced this anomaly by turning off the AP power of two APs, AP2 and AP4, deliberately for some time-slots. The *halt-period* of these two anomalous APs are not exactly the same, but has some overlaps. These two APs have different type of wireless users with different mobility patterns, but equal number of users. In AP2 users mostly remain connected to the same AP and have less freedom in movement (e.g. resembling the usage pattern of a classroom), while in AP4 users have more random movement (e.g. resembling the usage pattern of a public corridor).

Figure 6.7 and 6.8 demonstrate the ROC curves of anomaly detection results belong to AP2 and AP4, respectively. In these figures the detection sensitivity and specificity of three models are compared: HMM-UBM, SOHMMM, and Z-SOHMMM (SOHMMM with zero neighborhood). The results of this experiment show that the

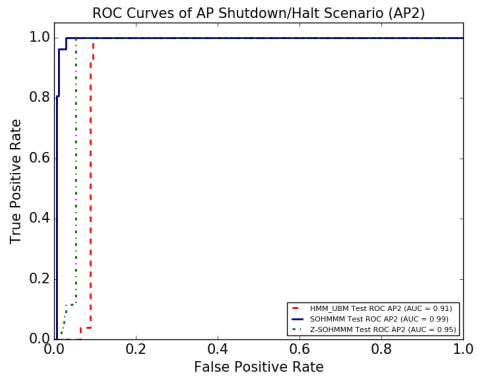


FIGURE 6.7: ROC curves of AP shutdown/halt Scenario (anomalous AP2).

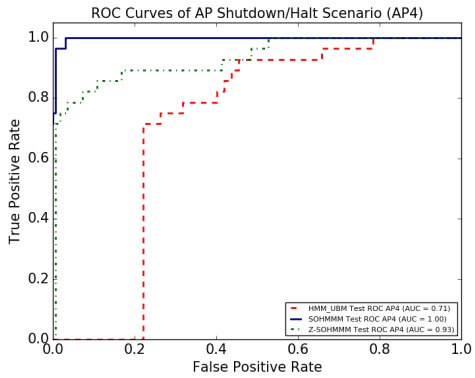


FIGURE 6.8: ROC curves of AP shutdown/halt Scenario (anomalous AP4).

SOHMMM algorithm outperformed the HMM-UBM and Z-SOHMMM techniques in both examples.

In both anomalous APs, SOHMMM achieved higher true positive rate (TPR) rather than the other two approaches, and only produced minor false positive rate (FPR), thus the area under curve (AUC) for both APs are very high (0.99 and 1.0). The Z-SOHMMM detection results, however, are slightly beneath the SOHMMM (0.95 and 0.93), and the HMM-UBM results are in the third place (0.91 and 0.71). Despite the fact that detection of this type of anomaly in AP4 is more challenging due to the mobility pattern of its users, SOHMMM achieved similar performance as in AP2 while the other two approaches performed inadequately.

AP Overload

In this experiment we simulated AP heavy usage caused by half of the users of AP2 (10 out of 20 users) during the *burst-period* for a number of time-slots. To resemble the heavy downloads of the wireless users, a burst server sends UDP packets in bursts, to the IP addresses of mobile users associated with AP2, during the *burst-period*. In the time of *sleep-period* the burst flow stops and the channel utilization gets back to normal.

In this simulation scenario, AP2 is the anomalous AP and the rest of the APs are supposed to be in a normal condition. Figure 6.9 depicts the ROC curves regarding the anomalous AP2. As this figure shows all the aforementioned models - HMM-UBM, SOHMMM, and Z-SOHMMM - performed quite well, basically close to perfect in terms of high TPR and low FPR. However, SOHMMM still outperformed the other two models and achieved ideal AUC (1.0).

Noise

In the current experiment we changed the level of noise power of AP2 by adjusting the value of *IsotropicBackgroundNoise* parameter in the simulator during the *noisy-period*. The default value of the background noise parameter is set to -110dBm in the simulator which is the minimum noise level in Wi-Fi networks 802.11 variants. In this anomalous scenario we increased the noise power to -95 dBm. This noise power only affects the users of AP2.

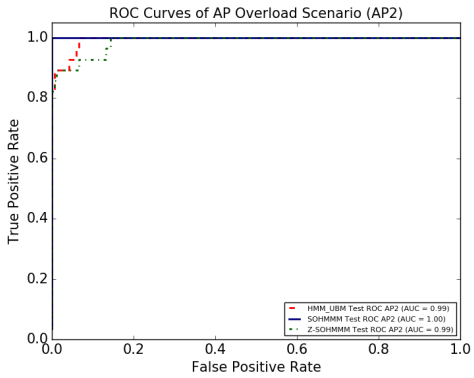


FIGURE 6.9: ROC curves of AP overload scenario (anomalous AP2).

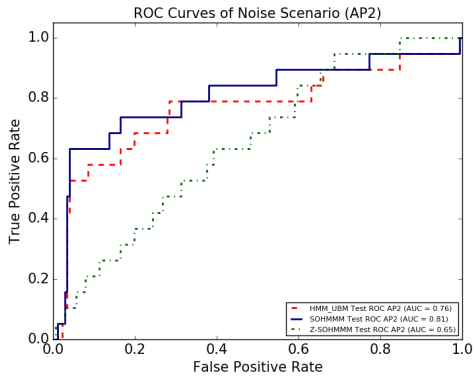


FIGURE 6.10: ROC curves of noise scenario (anomalous AP2).

Figure 6.10 displays the ROC curves related to this anomalous scenario depicted in AP2. As the AUC values demonstrate, in this example, the TPR versus FPR outcome could not achieve a close to perfect result similar to the previous anomalous cases. However the highest detection rate is obtained by the SOHMMM algorithm (AUC=0.81). The SOHMMM algorithm improved the basic detection results of HMM-UBM (AUC=0.76), while updating no neighbour in Z-SOHMMM approach got inferior results (AUC=0.65).

Flash Crowd

In wireless networks an unexpected surge of traffic occurs mostly due to the beginning or ending of an event when the majority of the wireless users abruptly enter or leave a place and consequently associate to or disassociate from an AP. We simulated this setting in two different cases:

- Arrival: simultaneous association of 10 new nodes to AP2.
- Departure: simultaneous dis-association of 10 existing nodes from AP2.

Figure 6.11 and 6.12 present the ROC curves of flash crowd anomaly regarding the arrival and departure scenarios, respectively. In arrival scenario in Figure 6.11, the SOHMMM algorithm achieved the highest AUC value (0.89), while the other two approaches, HMM-UBM and Z-SOHMMM, obtained almost identical AUC values (0.80), and lower than the SOHMMMM outcome. However, in departure scenario in Figure 6.12, the AUC values of the three approaches were very close to each other, while SOHMMM and Z-SOHMMM still achieved slightly higher value (0.97 in both models) rather than HMM-UBM model (0.96).

Miscellaneous Anomalies

In this experiment we simulated a number of anomalous cases together in one day data. The purpose of this experiment is to verify whether the proposed methodologies are capable of detecting a combination of different kinds of anomalies. It should be taken into consideration that anomalous periods of these different anomalies do not necessarily overlap with each other. In this scenario two adjacent APs are considered to be anomalous: AP2 and AP3. The wireless users of these two APs have

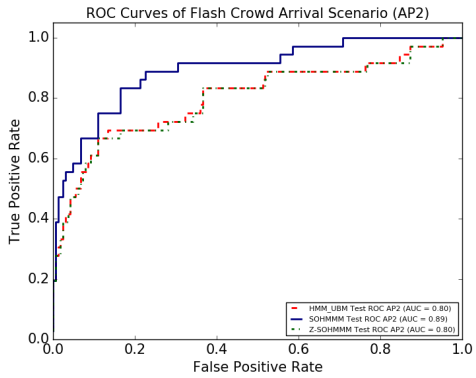


FIGURE 6.11: ROC curves of flash crowd arrival scenario (anomalous AP2).

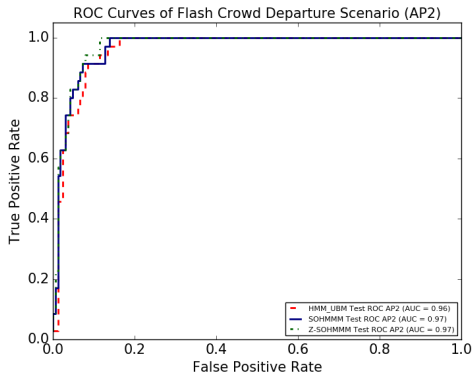


FIGURE 6.12: ROC curves of flash crowd departure scenario (anomalous AP2).

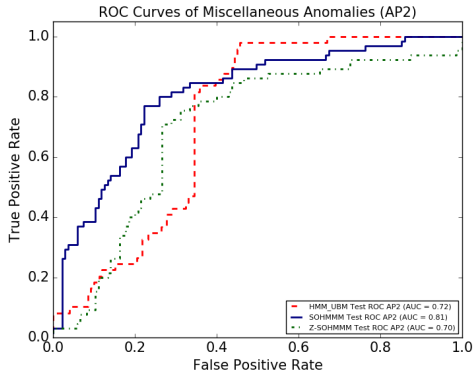


FIGURE 6.13: ROC curves of miscellaneous anomalies scenario (anomalous AP1).

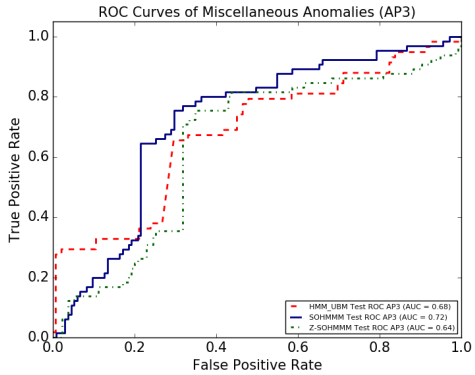


FIGURE 6.14: ROC curves of miscellaneous anomalies scenario (anomalous AP3).

alike mobility patterns and mostly remain connected to their initial AP (e.g. resembling a classroom usage pattern), however the number of users in AP2 is twice the number of users in AP3. The anomalies are sequentially ordered in these APs as follows:

- $AP2 : APShutdown/Halt + APOverload + Noise$
- $AP3 : APShutdown/Halt + Noise + APOverload$

The intervals of two anomalous cases consist of normal interactions between wireless users and APs. The duration of anomalous scenarios are different from each other. Figure 6.13 and 6.14 demonstrate the ROC curves of anomaly detection results regarding these two anomalous APs. In both cases the highest AUC value is achieved by SOHMMM algorithm (0.81 for AP2 and 0.72 for AP3). In the following paragraphs we analyze more profoundly the detection results of different anomalous cases in AP2 as an example.

Figure 6.15 displays the obtained likelihood results by HMM-UBM and SOHMMM models, respectively. The results from the Z-SOHMMM algorithm are similar to

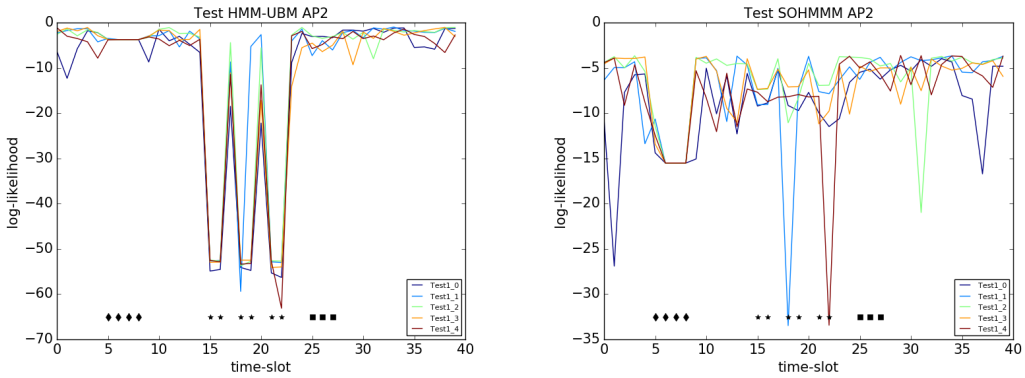


FIGURE 6.15: The log-likelihood series and actual anomalous points of miscellaneous anomalies scenario in AP2. *diamond*: AP shutdown/halt, *star*: AP overload, and *square*: noise. Left) HMM-UBM, right) SOHMMM.

SOHMMM, therefore to avoid repetition we only compare the recognition results of HMM-UBM and SOHMMM. The valley shapes in these figures show the sudden drops of the likelihood values during the anomalous periods. The black markers display the actual anomalous points generated during the simulation, *diamond* for AP shutdown/halt, *star* for AP overload, and *square* for noise anomalies.

Having examined the detected anomalies more carefully, it is observed that SOHMMM is able to detect AP shutdown/halt and AP overload anomalous points more accurately than the noise anomaly. While HMM-UBM is fitted to AP overload anomalous case and is not able to detect nor AP shutdown/halt neither noise anomaly. As the outcome of Section 6.5.2 also shows, the detection of noise anomalous case is not that straightforward. However, recognition of AP shutdown/halt in addition to AP overload anomaly by SOHMMM is a significant result, because these two types of anomalies are very different in nature. AP shutdown/halt is a silent period where there is minimum or no activity recorded, while AP overload is high utilization of the network by wireless users. Detection of these two contrasting anomalies in the same observation sequence by SOHMMM, while HMM-UBM only reacted to AP overload anomaly, show that SOHMMM improves the HMM-UBM model and plays an important role in anomaly detection scheme.

The experimental results show that including spatial information of the neighboring APs in the SOHMMM algorithm improve the anomaly detection results in various simulated anomalous cases. The reason is that in SOHMMM, every time a new observation sequence is arrived, not only the winner HMM, but also the nearby HMMs activate each other to gain some information from the new observation. Hence, the HMM models learn from the usage pattern of their neighbors in addition to their own. The spatial intuition that supports SOHMMM improved the anomaly detection results in AP usage data, because in the real scenario the usage pattern of the neighboring APs affect each other. For instance when an AP stops working, the wireless users that were already connected to that AP connect to other APs in the neighborhood, and this might have some influences on the usage pattern of the nearby APs.

6.6 Conclusion

In this chapter we applied a hybrid integration of the Self-Organizing Map (SOM) and the Hidden Markov Model (HMM), called SOHMMM for anomaly detection in 802.11 wireless network. We further extended the online gradient descent unsupervised learning algorithm of SOHMMM for multivariate Gaussian emissions. We employed this algorithm specifically for anomaly detection in 802.11 wireless AP usage data.

Experimental analysis consists of two main parts: *Synthetic Data*, and *Wireless Simulation Data*. In synthetic data analysis we generated six random HMMs and trained them with the observation sequences of the two reference HMMs with predefined parameters. We then estimated the distance between HMMs as Monte Carlo approximation of the Kullback-Leibler divergence, and computed the final HMM clusters based on the minimum distance between random HMMs and reference HMMs. We repeated the experiment for various observation sequence lengths and different neighborhood sizes, and showed that the SOHMMM algorithm with the large neighborhood and the longest observation sequence provides better estimation of the reference models including all the HMMs in the vicinity. Moreover, we presented the decay of the learning rate and the convergence of the loss function.

In *Wireless Simulation Data* analysis, we showed how the SOHMMM algorithm improved the anomaly detection accuracy and sensitivity compared to HMM-UBM and Z-SOHMMM techniques in AP shutdown/halt, AP overload, noise, and flash crowd anomalous scenarios. We further investigated the combination of several anomalies in one observation sequence as miscellaneous anomalies and showed that SOHMMM is capable of detecting contrasting anomalous cases while HMM-UBM is not.

Chapter 7

Conclusions and Future Work

In 802.11 Wireless Networks, detecting faulty equipment, poor radio conditions, and changes in user behavior through anomaly detection is of great importance for network managers. The traffic load and users movement on different access points (APs) in a wireless covered area vary from time to time, making these network management tasks harder. AP usage modeling and anomaly detection in hotspots would assist network administrators to ensure long-term quality of service by analyzing various connectivity factors of wireless users in particular localities. One of the objectives of this work is to inspect and characterize the usage pattern of the wireless networks and its inherent dynamics by exploring the spatial proximity of access points and their timely usage pattern simultaneously and to provide robust models for anomaly detection.

We proposed an automatic diagnostic tool that analyzes the usage data of the APs collected from a RADIUS authentication server, a similar but smaller Testbed, and a middle-sized wireless simulation network. The large log data of RADIUS authentication is collected from FEUP in approximately two years, and contains the connection summary of more than 45 thousands users associated to 364 APs. The small exploratory Testbed is deployed in FreeRADIUS server and consists of one AP and six wireless users. The wireless simulation is performed in infrastructure mode in OMNeT++/INET and consists of 10 APs and 100 wireless stations. The FEUP data set is conducted on a real university campus, however do not contain ground truth of anomalous events. The Testbed deployment and wireless simulation data set contain ground truth of a number of anomalous cases generated in them deliberately, however they are reproduced in smaller scales.

We applied probabilistic learning algorithms, specifically Hidden Markov Model and its variations, to build a single model for all APs, individual models for each AP, and several models for group of APs, and identified anomalous events with a margin of certitude. We further presented several indicators of outliers by HMM parameters' analysis, and provided a number of anomalous patterns associated with such networks in terms of HMM state transitions. The results showed that HMM models were able to discover a portion of the state of the art's outliers (univariate, multivariate and temporal). The HMM models also introduced some additional outliers that could be justified by HMM parameters indicators (large distance to assigned HMM state, low transition probability, and rare state modification). The single and mixture models outperformed the individual HMMs in terms of accuracy and HMM indicators conformity.

We extended the analysis by proposing time-invariant (GMM) and time-variant (HMM) modeling approaches and utilizing these models for anomaly detection. We conducted an experiment as a case study on FEUP data set, and examined the goodness of fit using log-likelihood values. Furthermore, we described the anomaly detection techniques by GMM and HMM, and explored the addressed methodologies

to detect anomalies at the same experimental data of the previous case study. We justified the detected anomalous points in absence of the ground truth in the large data set, and measured the false positive rate (FPR), true negative rate (TNR), true positive rate (TPR), accuracy (ACC) and F1 score in normal and anomalous test data belong to Testbed data set. The experimental results demonstrated that HMM outperformed GMM in obtaining higher detection ratio while producing minor false alarm.

Moreover, we proposed Universal Background Modeling (UBM) approach for a robust initialization of the HMM models. The experimental results, performed on wireless simulation network, showed that HMM and HMM-UBM models are both capable of detecting a considerable portion of anomalies while producing only a small false positive ratio. This is promising for in HMM-UBM model all APs' data, regardless of being normal or containing anomalous events, is utilized to initialize the HMM model. Thus, in unsupervised learning, where the normal data is not known beforehand, HMM-UBM yields a robust model, as reliable as HMM initialized with normal data. The anomaly detection results are compared to baseline approaches, RawData and PCA, and showed that HMM is required to detect such sort of anomalies while the simpler baseline approaches are inadequate for anomaly detection.

We applied a hybrid integration of the Self-Organizing Map (SOM) and the Hidden Markov Model (HMM), called SOHMMM for anomaly detection in 802.11 wireless network. The Self-Organizing Hidden Markov Model Map (SOHMMM) deals with the spatial connection of HMMs along with the inherent temporal dependency of data sequences. We further extended the online gradient descent unsupervised learning algorithm of SOHMMM for multivariate Gaussian emissions. Experimental analysis consist of two main parts: synthetic data, and wireless simulation data. In synthetic data analysis we generated six random HMMs and trained them with observation sequences of two reference HMMs with predefined parameters. We then estimated the distance between HMMs as Monte Carlo approximation of the Kullback-Leibler divergence, and computed the final HMM clusters based on the minimum distance between random HMMs and reference HMMs. We repeated the experiment for various observation sequence lengths and different neighborhood sizes, and showed that the SOHMMM algorithm with the large neighborhood and the longest observation sequence provides better estimation of the reference models including all the HMMs in the vicinity. Further, we presented the decay of the learning rate and the convergence of the loss function. In wireless simulation data analysis, we showed how the SOHMMM algorithm improve the anomaly detection accuracy and sensitivity compared to HMM-UBM and Z-SOHMMM (SOHMMM with zero neighborhood) techniques in AP shutdown/halt, AP overload, noise, and flash crowd anomalous scenarios. We further investigated the combination of several anomalies in one observation sequence as miscellaneous anomalies and we showed that SOHMMM is capable of detecting contrasting anomalous cases while HMM-UBM is not.

Overall, all our investigations in this thesis are original and to the best of our knowledge could not have been done before our contributions. In future work, we plan to explore anomalous pattern recognition techniques to distinguish between various anomalous scenarios. We intend to focus on micro modeling approaches based on HMMs to characterize the sub-sequences of time-series and distinguish the anomalous cases. In this regard, we also intend to fully develop our previously proposed unsupervised anomaly detection algorithm in Section 5.3.2, to detect anomalous periods and distinguish between various anomalous types in unlabeled data

sets. We also intend to explore other types of anomalous patterns such as intrusion, and rough APs among others. Furthermore, we plan to apply the SOHMMM algorithm on our large data set (in Section 3.2) and analyze the impact of physical and logical proximity of APs in spatial connection of HMMs. The logical proximity can be estimated according to the ping-pong algorithm proposed in [21]. Further analysis regarding the classification of APs in a large network are among the compelling future lines of research. Such classification efforts arise the question of usage similarity or difference in formed categories with different characteristics. Lastly, if we had a large network data set and a network manager willing to annotate the data and provide us with some expert intuition in terms of anomalous, suspicious, or normal type of events, we would like to apply the proposed techniques on such data and evaluate the presented algorithms in real scenarios.

Chapter 8

Appendix A

Lemma 1.

$$\frac{\partial P(O|\lambda)}{\partial a_{rs}} = \sum_{l=1}^{T-1} \left[\alpha_l(r) b_s(o_{l+1}) \beta_{l+1}(s). \right. \tag{8.1}$$

Proposition 1.

$$\frac{\partial P(O|\lambda)}{\partial w_{ij}} = a_{ij} \sum_{l=1}^{T-1} \left[\alpha_l(i) b_j(o_{l+1}) \beta_{l+1}(j) - \alpha_l(i) \beta_l(i)\right]. \tag{8.2}$$

Lemma 2.

$$\frac{\partial P(O|\lambda)}{\partial b_x(y)} = \frac{1}{b_x(y)} \sum_{l=1}^{T} I\{o_l = y|\lambda\} \alpha_l(x) \beta_l(x). \tag{8.3}$$

Proposition 2.

$$\frac{\partial P(O|\lambda)}{\partial r_{jt}} = \sum_{l=1}^{T} \left[I\{o_l = t|\lambda\} \alpha_l(j) \beta_l(j) - b_j(t) \alpha_l(j) \beta_l(j)\right]. \tag{8.4}$$

Lemma 3.

$$\frac{\partial P(O|\lambda)}{\partial \pi_r} = b_r(o_1 \beta_1(r)). \tag{8.5}$$

Proposition 3.

$$\frac{\partial P(O|\lambda)}{\partial u_j} = \pi_j b_j(o_1) \beta_1(j) - \pi_j P(O|\lambda). \tag{8.6}$$

The proof of Lemma 1 can be found in [136]. And the proof of Lemma 2 and Proposition 2 can be found in [80].

Bibliography

[1] *The Internet Engineering Task Force (IETF)*. https://www.ietf.org/. Accessed in May 2019.

[2] *Remote Authentication Dial In User Service (RADIUS)*. http://tools.ietf.org/html/rfc2865. Accessed in May 2019.

[3] *RADIUS Accounting*. http://tools.ietf.org/html/rfc2866. Accessed in May 2019.

[4] Anthony J Nicholson et al. "Improved access point selection". In: *Proceedings of the 4th international conference on Mobile systems, applications and services*. ACM. 2006, pp. 233–245.

[5] Martin Heusse et al. "Performance anomaly of 802.11 b". In: *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*. Vol. 2. IEEE. 2003, pp. 836–843.

[6] Diego Dujovne, Thierry Turletti, and Fethi Filali. "A taxonomy of IEEE 802.11 wireless parameters and open source measurement tools". In: *IEEE Communications Surveys & Tutorials* 12.2 (2010), pp. 249–262.

[7] Atul Adya et al. "Architecture and techniques for diagnosing faults in IEEE 802.11 infrastructure networks". In: *Proceedings of the 10th annual international conference on Mobile computing and networking*. ACM. 2004, pp. 30–44.

[8] Yu-Chung Cheng et al. *Jigsaw: Solving the puzzle of enterprise 802.11 analysis*. Vol. 36. 4. ACM, 2006, pp. 39–50.

[9] Utpal Paul et al. "Passive measurement of interference in wifi networks with application in misbehavior detection". In: *IEEE transactions on mobile computing* 12.3 (2013), pp. 434–446.

[10] Anisa Allahdadi and Ricardo Morla. "Anomaly detection and modeling in 802.11 wireless networks". In: *Journal of Network and Systems Management* 27.1 (2019), pp. 3–38.

[11] Anisa Allahdadi, Ricardo Morla, and Jaime S Cardoso. "802.11 Wireless Simulation and Anomaly Detection using HMM and UBM". In: *arXiv preprint arXiv:1707.02933* (2018).

[12] Anisa Allahdadi, Ricardo Morla, and Jaime S Cardoso. "Outlier detection in 802.11 wireless access points using Hidden Markov Models". In: *2014 7th IFIP Wireless and Mobile Networking Conference (WMNC)*. IEEE. 2014, pp. 1–8.

[13] Anisa Allahdadi et al. "Predicting short 802.11 sessions from radius usage data". In: *38th Annual IEEE Conference on Local Computer Networks-Workshops*. IEEE. 2013, pp. 1–8.

[14] Diane Tang and Mary Baker. "Analysis of a local-area wireless network". In: *Proceedings of the 6th annual international conference on Mobile computing and networking*. ACM. 2000, pp. 1–10.

[15] Diane Tang and Mary Baker. "Analysis of a metropolitan-area wireless network". In: *Wireless Networks* 8.2/3 (2002), pp. 107–120.

[16] David Kotz and Kobby Essien. "Analysis of a campus-wide wireless network". In: *Wireless Networks* 11.1-2 (2005), pp. 115–133.

[17] Tristan Henderson, David Kotz, and Ilya Abyzov. "The changing usage of a mature campus-wide wireless network". In: *Proceedings of the 10th annual international conference on Mobile computing and networking*. ACM. 2004, pp. 187–201.

[18] Magdalena Balazinska and Paul Castro. "Characterizing mobility and network usage in a corporate wireless local-area network". In: *Proceedings of the 1st international conference on Mobile systems, applications and services*. ACM. 2003, pp. 303–316.

[19] Minkyong Kim and David Kotz. "Periodic properties of user mobility and access-point popularity". In: *Personal and Ubiquitous Computing* 11.6 (2007), pp. 465–479.

[20] Mathias Boc, Anne Fladenmuller, and Marcelo Dias De Amorim. "Towards self-characterization of user mobility patterns". In: *Mobile and Wireless Communications Summit, 2007. 16th IST*. IEEE. 2007, pp. 1–5.

[21] Ricardo Sousa et al. "Analysis of the logical proximity between 802.11 access points". In: *CRC 2012: 12ª Conferência sobre Redes de Computadores*. 2013, pp. 47–55.

[22] Libo Song et al. "Evaluating next-cell predictors with extensive Wi-Fi mobility data". In: *IEEE transactions on mobile computing* 5.12 (2006), pp. 1633–1649.

[23] Yohan Chon et al. "Evaluating mobility models for temporal prediction with high-granularity mobility data". In: *Pervasive computing and communications (PerCom), 2012 IEEE international conference on*. IEEE. 2012, pp. 206–212.

[24] Wei-jen Hsu and Ahmed Helmy. "On nodal encounter patterns in wireless LAN traces". In: *IEEE Transactions on Mobile Computing* 9.11 (2010), pp. 1563–1577.

[25] Wei-jen Hsu, Debojyoti Dutta, and Ahmed Helmy. "Mining behavioral groups in large wireless LANs". In: *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*. ACM. 2007, pp. 338–341.

[26] Wei-jen Hsu, Debojyoti Dutta, and Ahmed Helmy. "Structural analysis of user association patterns in university campus wireless lans". In: *IEEE Transactions on Mobile Computing* 11.11 (2012), pp. 1734–1748.

[27] Yung-Chih Chen et al. "Group detection in mobility traces". In: *Proceedings of the 6th international wireless communications and mobile computing conference*. ACM. 2010, pp. 875–879.

[28] Matthew J Williams, Roger M Whitaker, and Stuart M Allen. "Measuring individual regularity in human visiting patterns". In: *Privacy, Security, Risk and Trust (PASSAT), 2012 international conference on and 2012 international confernece on Social Computing (SocialCom)*. IEEE. 2012, pp. 117–122.

[29] Emery N Brown, Robert E Kass, and Partha P Mitra. "Multiple neural spike train data analysis: state-of-the-art and future challenges". In: *Nature neuroscience* 7.5 (2004), p. 456.

[30] Maria Papadopouli, Haipeng Shen, and Manolis Spanakis. "Characterizing the duration and association patterns of wireless access in a campus". In: *Wireless Conference 2005-Next Generation Wireless and Mobile Communications and Services (European Wireless), 11th European*. VDE. 2005, pp. 1–7.

[31] Caleb Phillips and Suresh Singh. "An empirical activity model for wlan users". In: *IEEE INFOCOM 2008-The 27th Conference on Computer Communications*. IEEE. 2008, pp. 2065–2073.

[32] Enrica Zola and Francisco Barcelo-Arroyo. "Impact of mobility models on the cell residence time in WLAN networks". In: *Sarnoff Symposium, 2009. SARNOFF'09. IEEE*. IEEE. 2009, pp. 1–5.

[33] Maria Papadopouli et al. "Short-term traffic forecasting in a campus-wide wireless network". In: *Personal, Indoor and Mobile Radio Communications, 2005. PIMRC 2005. IEEE 16th International Symposium on*. Vol. 3. IEEE. 2005, pp. 1446–1452.

[34] Parag Kulkarni, Tim Lewis, and Zhong Fan. "Simple traffic prediction mechanism and its applications in wireless networks". In: *Wireless Personal Communications* 59.2 (2011), pp. 261–274.

[35] Edward Chlebus and Gautam Divgi. "The Pareto or truncated Pareto distribution? Measurement-based modeling of session traffic for Wi-Fi wireless Internet access". In: *Wireless Communications and Networking Conference, 2007. WCNC 2007. IEEE*. IEEE. 2007, pp. 3625–3630.

[36] Amitabha Ghosh et al. "Modeling and characterization of large-scale Wi-Fi traffic in public hot-spots". In: *2011 Proceedings IEEE INFOCOM*. IEEE. 2011, pp. 2921–2929.

[37] Félix Hernández-Campos et al. "Spatio-temporal modeling of traffic workload in a campus WLAN". In: *Proceedings of the 2nd annual international workshop on Wireless internet*. ACM. 2006, p. 1.

[38] Karolina Baras and Adriano Moreira. "Symbolic space modeling based on WiFi network data analysis". In: *2010 Seventh International Conference on Networked Sensing Systems (INSS)*. IEEE. 2010, pp. 273–276.

[39] Dossa Massa and Ricardo Morla. "Modeling 802.11 AP usage through daily keep-alive event counts". In: *Wireless networks* 19.5 (2013), pp. 1005–1022.

[40] Karolina Baras and Adriano Moreira. "Anomaly detection in university campus WiFi zones". In: *2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*. IEEE. 2010, pp. 202–207.

[41] Ramya Raghavendra et al. "Unwanted link layer traffic in large IEEE 802.11 wireless networks". In: *IEEE Transactions on Mobile Computing* 9.9 (2010), pp. 1212–1225.

[42] Prashanth Aravinda Kumar Acharya et al. "Congestion-aware rate adaptation in wireless networks: A measurement-driven approach". In: *Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON'08. 5th Annual IEEE Communications Society Conference on*. IEEE. 2008, pp. 1–9.

[43] Rohan Murty et al. "Designing High Performance Enterprise Wi-Fi Networks." In: *NSDI*. Vol. 8. 2008, pp. 73–88.

[44] Hongseok Kim et al. "Distributed α-optimal user association and cell load balancing in wireless networks". In: *IEEE/ACM Transactions on Networking (TON)* 20.1 (2012), pp. 177–190.

[45] HJ Pan and Srinivasan Keshav. "Detection and repair of faulty access points". In: *IEEE Wireless Communications and Networking Conference, 2006. WCNC 2006.* Vol. 1. IEEE. 2006, pp. 532–538.

[46] Suman Jana and Sneha K Kasera. "On fast and accurate detection of unauthorized wireless access points using clock skews". In: *IEEE Transactions on Mobile Computing* 9.3 (2010), pp. 449–462.

[47] Atul Adya et al. "Architecture and techniques for diagnosing faults in IEEE 802.11 infrastructure networks". In: *Proceedings of the 10th annual international conference on Mobile computing and networking*. ACM. 2004, pp. 30–44.

[48] Aditya Akella et al. "Self-management in chaotic wireless deployments". In: *Wireless Networks* 13.6 (2007), pp. 737–755.

[49] Ioannis Broustis et al. "Measurement-driven guidelines for 802.11 WLAN design". In: *IEEE/ACM Transactions on Networking (TON)* 18.3 (2010), pp. 722–735.

[50] Ramakrishna Gummadi et al. "Understanding and mitigating the impact of RF interference on 802.11 networks". In: *ACM SIGCOMM Computer Communication Review* 37.4 (2007), pp. 385–396.

[51] Dossa Massa and Ricardo Morla. "Abrupt ending of 802.11 ap connections". In: *Computers and Communications (ISCC), 2013 IEEE Symposium on*. IEEE. 2013, pp. 000348–000353.

[52] Vivek Shrivastava et al. "PIE in the Sky: Online Passive Interference Estimation for Enterprise WLANs." In: *NSDI*. Vol. 11. 2011, pp. 25–25.

[53] Anmol Sheth et al. "MOJO: A distributed physical layer anomaly detection system for 802.11 WLANs". In: *Proceedings of the 4th international conference on Mobile systems, applications and services*. ACM. 2006, pp. 191–204.

[54] Kaushik Lakshminarayanan, Srinivasan Seshan, and Peter Steenkiste. "Understanding 802.11 performance in heterogeneous environments". In: *Proceedings of the 2nd ACM SIGCOMM workshop on Home networks*. ACM. 2011, pp. 43–48.

[55] Ratul Mahajan et al. "Analyzing the MAC-level behavior of wireless networks in the wild". In: *ACM SIGCOMM Computer Communication Review*. Vol. 36. 4. ACM. 2006, pp. 75–86.

[56] Syed A Khayam and Hayder Radha. "Markov-based modeling of wireless local area networks". In: *Proceedings of the 6th ACM international workshop on Modeling analysis and simulation of wireless and mobile systems*. ACM. 2003, pp. 100–107.

[57] Ankur Kamthe, Miguel A Carreira-Perpinán, and Alberto E Cerpa. "M&M: multi-level Markov model for wireless link simulations". In: *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*. ACM. 2009, pp. 57–70.

[58] Chittabrata Ghosh et al. "Markov chain existence and hidden Markov models in spectrum sensing". In: *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*. IEEE. 2009, pp. 1–6.

[59] Yuanyuan Qiao et al. "Characterizing flow, application, and user behavior in mobile networks: A framework for mobile big data". In: *IEEE Wireless Communications* 25.1 (2018), pp. 40–49.

[60] Hirotaka Kawazu et al. "Analytical method of web user behavior using Hidden Markov Model". In: *2016 IEEE International Conference on Big Data (Big Data)*. IEEE. 2016, pp. 2518–2524.

[61] Shanshan Tu et al. "HMM-based User Behavior Prediction Method in Heterogeneous Cellular Networks." In: *International Journal of Performability Engineering* 14.9 (2018).

[62] Wojciech Bednarczyk and Piotr Gajewski. "Hidden Markov Models Based Channel Status Prediction for Cognitive Radio Networks". In: *Session 4P6 RF and Wireless Communication* (July 2015), p. 2088.

[63] Ihsan Akbar, William H Tranter, et al. "Dynamic spectrum allocation in cognitive radio using hidden Markov models: Poisson distributed case". In: *SoutheastCon, 2007. Proceedings. IEEE*. IEEE. 2007, pp. 196–201.

[64] Chittabrata Ghosh et al. "Markov chain existence and hidden Markov models in spectrum sensing". In: *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*. IEEE. 2009, pp. 1–6.

[65] Vamsi Krishna Tumuluru, Ping Wang, and Dusit Niyato. "Channel status prediction for cognitive radio networks". In: *Wireless Communications and Mobile Computing* 12.10 (2012), pp. 862–874.

[66] Pratap S Prasad and Prathima Agrawal. "Movement prediction in wireless networks using mobility traces". In: *Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE*. IEEE. 2010, pp. 1–5.

[67] Ahlam Ben Cheikh et al. "Optimized handoff with mobility prediction scheme using hmm for femtocell networks". In: *Communications (ICC), 2015 IEEE International Conference on*. IEEE. 2015, pp. 3448–3453.

[68] D Alberto, P Antonio, SR Pierluigi, et al. "An HMM Approach to Internet Traffic Modeling". In: *proceeding of IEEE GLOBECOM*. 2006.

[69] Alberto Dainotti et al. "Classification of network traffic via packet-level hidden markov models". In: *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*. IEEE. 2008, pp. 1–5.

[70] José Everardo Bessa Maia and Raimir Holanda Filho. "Internet traffic classification using a Hidden Markov model". In: *Hybrid Intelligent Systems (HIS), 2010 10th International Conference on*. IEEE. 2010, pp. 37–42.

[71] Anand Kashyap, Utpal Paul, and Samir R Das. "Deconstructing interference relations in WiFi networks". In: *Sensor Mesh and Ad Hoc Communications and Networks (SECON), 2010 7th Annual IEEE Communications Society Conference on*. IEEE. 2010, pp. 1–9.

[72] Utpal Paul et al. "Passive measurement of interference in wifi networks with application in misbehavior detection". In: *IEEE transactions on mobile computing* 12.3 (2013), pp. 434–446.

[73] Rahul Khanna and Huaping Liu. "System approach to intrusion detection using hidden markov model". In: *Proceedings of the 2006 international conference on Wireless communications and mobile computing*. ACM. 2006, pp. 349–354.

[74] Chia-Mei Chen et al. "Anomaly network intrusion detection using hidden Markov model". In: *Int. J. Innov. Comput. Inform. Control* 12 (2016), pp. 569–580.

[75] Takashi Fuse and Keita Kamiya. "Statistical anomaly detection in human dynamics monitoring using a hierarchical dirichlet process hidden Markov model". In: *IEEE Transactions on Intelligent Transportation Systems* 18.11 (2017), pp. 3083–3092.

[76] Gen Niina and Hiroshi Dozono. "The spherical hidden markov self organizing map for learning time series data". In: *International Conference on Artificial Neural Networks*. Springer. 2012, pp. 563–570.

[77] Nobuhiko Yamaguchi. "Self-organizing hidden markov models". In: *International Conference on Neural Information Processing*. Springer. 2010, pp. 454–461.

[78] George Caridakis et al. "SOMM: Self organizing Markov map for gesture recognition". In: *Pattern Recognition Letters* 31.1 (2010), pp. 52–59.

[79] Rakia Jaziri et al. "SOS-HMM: self-organizing structure of hidden Markov model". In: *International Conference on Artificial Neural Networks*. Springer. 2011, pp. 87–94.

[80] Christos Ferles and Andreas Stafylopatis. "Self-organizing hidden markov model map (SOHMMM)". In: *Neural Networks* 48 (2013), pp. 133–147.

[81] Panu Somervuo. "Competing hidden markov models on the self-organizing map". In: *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*. Vol. 3. IEEE. 2000, pp. 169–174.

[82] Mikko Kurimo and Panu Somervuo. "Using the Self-Organizing Map to speed up the probability density estimation for speech recognition with mixture density HMMs". In: *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*. Vol. 1. IEEE. 1996, pp. 358–361.

[83] Christos Ferles, Georgios Siolas, and Andreas Stafylopatis. "Scaled self-organizing map–hidden Markov model architecture for biological sequence clustering". In: *Applied Artificial Intelligence* 27.6 (2013), pp. 461–495.

[84] Wael Khreich et al. "A survey of techniques for incremental learning of HMM parameters". In: *Information Sciences* 197 (2012), pp. 105–130.

[85] Sung-Bae Cho. "Incorporating soft computing techniques into a probabilistic intrusion detection system". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 32.2 (2002), pp. 154–160.

[86] Wei Wang et al. "Profiling program behavior for anomaly intrusion detection based on the transition and frequency property of computer audit data". In: *computers & security* 25.7 (2006), pp. 539–550.

[87] Jiancong Chen, S-HG Chan, and Soung-Chang Liew. "Mixed-mode WLAN: the integration of ad hoc mode with wireless LAN infrastructure". In: *Global Telecommunications Conference, 2003. GLOBECOM'03. IEEE*. Vol. 1. IEEE. 2003, pp. 231–235.

[88] *The Network Simulator - ns-2*. `https://www.isi.edu/nsnam/ns/`. accessed May 2019.

[89] Victor Kulgachev and Hetal Jasani. "802.11 networks performance evaluation using Opnet". In: *Proceedings of the 2010 ACM conference on Information technology education*. ACM. 2010, pp. 149–152.

[90] *ns-3 Network Simulator*. https://www.nsnam.org/. Accessed in May 2019.

[91] *OPNET Technologies*. https://www.riverbed.com/gb/products/steelcentral/opnet.html. Accessed in May 2019.

[92] *OMNeT++ Discrete Event Simulator*. https://www.omnetpp.org/. accessed May 2019.

[93] *INET Framework*. https://inet.omnetpp.org/. accessed in May 2019.

[94] Michael Bredel and Martin Bergner. "On the accuracy of ieee 802.11 g wireless lan simulations using omnet++". In: *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). 2009, p. 81.

[95] Mina Malekzadeh et al. "Validating reliability of OMNeT++ in wireless networks DoS attacks: simulation vs. testbed." In: *IJ Network Security* 13.1 (2011), pp. 13–21.

[96] Riyadh Qashi, Martin Bogdan, and Klaus Haenssgen. "Analysis of packet throughput and delay in IEEE 802.11 WLANs with UDP traffic". In: *International Conference on Mobile Communications, Networking and Applications (MobiCONA). Proceedings*. Global Science and Technology Forum. 2011, p. M48.

[97] Steve Woon, Eric Wu, and Ahmet Sekercioglu. "A simulation model of IEEE802.11b for performance analysis of wireless LAN protocols". In: *Australian Telecommunications, Networks and Applications Conference (ATNAC)*. Vol. 162. 2003.

[98] Miquel van Smoorenburg and Alan DeKok. *The FreeRADIUS Project*. http://freeradius.org/. Accessed in May 2019.

[99] Dan McInerney. *Wifijammer*. https://github.com/DanMcInerney/wifijammer. Accessed in May 2019.

[100] Bo Fu et al. "Wireless background noise in the Wi-Fi spectrum". In: *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM'08. 4th International Conference on*. IEEE, 2008, pp. 1–7.

[101] Arpit Gupta, Jeongki Min, and Injong Rhee. "WiFox: Scaling WiFi performance for large audience environments". In: *Proceedings of the 8th international conference on Emerging networking experiments and technologies*. ACM. 2012, pp. 217–228.

[102] Atul Adya et al. "Architecture and techniques for diagnosing faults in IEEE 802.11 infrastructure networks". In: *Proceedings of the 10th annual international conference on Mobile computing and networking*. ACM. 2004, pp. 30–44.

[103] Fan Zhang et al. "Inferring users' online activities through traffic analysis". In: *Proceedings of the fourth ACM conference on Wireless network security*. ACM. 2011, pp. 59–70.

[104] Charles V Wright, Fabian Monrose, and Gerald M Masson. "On inferring application protocol behaviors in encrypted network traffic". In: *Journal of Machine Learning Research* 7.Dec (2006), pp. 2745–2769.

[105] Lawrence R Rabiner and Biing-Hwang Juang. "An introduction to hidden Markov models". In: *ieee assp magazine* 3.1 (1986), pp. 4–16.

[106] Lawrence R Rabiner. "A tutorial on hidden Markov models and selected applications in speech recognition". In: *Proceedings of the IEEE* 77.2 (1989), pp. 257–286.

[107] Geoffrey McLachlan and David Peel. "ML fitting of mixture models". In: *Finite Mixture Models (Hoboken, NJ: John Wiley and Sons, Inc.)* (2005), pp. 40–80.

[108] Weiyu Zhang, Qingbo Yang, and Yushui Geng. "A survey of anomaly detection methods in networks". In: *2009 International Symposium on Computer Network and Multimedia Technology*. IEEE. 2009, pp. 1–3.

[109] *Box Plot Statistics*. https://stat.ethz.ch/R-manual/R-devel/library/grDevices/html/boxplot.stats.html. accessed May 2019.

[110] *Mahalanobis Distance*. https://stat.ethz.ch/R-manual/R-devel/library/stats/html/mahalanobis.html. accessed May 2019.

[111] *Adjusted Quantile Plot*. https://www.rdocumentation.org/packages/mvoutlier/versions/2.0.9/topics/aq.plot. accessed May 2019.

[112] P. Filzmoser and M. Gschwandtner. *mvoutlier: Multivariate outlier detection based on robust methods*. accessed May 2019. 2013. URL: http://CRAN.R-project.org/package=mvoutlier.

[113] Manish Gupta et al. "Outlier detection for temporal data: A survey". In: *IEEE Transactions on Knowledge and Data Engineering* 26.9 (2014), pp. 2250–2267.

[114] R Andrew Weekley, Robert K Goodrich, and Larry B Cornman. "An algorithm for classification and outlier detection of time-series data". In: *Journal of Atmospheric and Oceanic Technology* 27.1 (2010), pp. 94–107.

[115] Fabrizio Angiulli and Fabio Fassetti. "Detecting Distance-based Outliers in Streams of Data". In: *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*. CIKM '07. Lisbon, Portugal: ACM, 2007, pp. 811–820.

[116] *Seasonal Decomposition of Time Series by Loess*. https://www.rdocumentation.org/packages/stats/versions/3.5.3/topics/stl. accessed May 2019.

[117] *Loess Smoothing*. https://www.statisticshowto.datasciencecentral.com/lowess-smoothing/. accessed May 2019.

[118] *Bhattacharyya Distance Between Gaussian Distributions*. https://www.rdocumentation.org/packages/fpc/versions/2.1-11.1/topics/bhattacharyya.dist. accessed May 2019.

[119] Claude Sammut and Geoffrey I Webb. *Ensemble Learning*. Springer Science & Business Media, 2011, pp. 312–320.

[120] Douglas Reynolds. "Gaussian mixture models". In: *Encyclopedia of Biometrics* (2015), pp. 827–832.

[121] Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn. "Speaker verification using adapted Gaussian mixture models". In: *Digital signal processing* 10.1-3 (2000), pp. 19–41.

[122] Chris Fraley and Adrian E Raftery. "Model-based clustering, discriminant analysis, and density estimation". In: *Journal of the American statistical Association* 97.458 (2002), pp. 611–631.

[123] Chris Fraley et al. *mclust Version 5.1 for R: Normal Mixture Modeling for Model-Based Clustering, Classification, and Density Estimation*. 597. Technical Report. 2015.

[124] A. Schliep et al. *GHMM Library*. http://ghmm.org. Accessed in May 2019.

[125] *System Sciences at Isis*. http://systems-sciences.uni-graz.at/etextbook/bigdata/confusionmatrix.html. Accessed in May 2019.

[126] Douglas Reynolds. "Universal Background Models". In: *Encyclopedia of Biometrics*. Ed. by Stan Z. Li and Anil K. Jain. Boston, MA: Springer US, 2015, pp. 1547–1550.

[127] Manish Gupta et al. "Outlier detection for temporal data: A survey". In: *IEEE Transactions on Knowledge and Data Engineering* 26.9 (2014), pp. 2250–2267.

[128] Dong Zhang et al. "Semi-supervised adapted hmms for unusual event detection". In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 1. IEEE. 2005, pp. 611–618.

[129] Pierre Baldi and Yves Chauvin. "Smooth on-line learning algorithms for hidden Markov models". In: *Neural Computation* 6.2 (1994), pp. 307–318.

[130] Teuvo Kohonen. "Self-organized formation of topologically correct feature maps". In: *Biological cybernetics* 43.1 (1982), pp. 59–69.

[131] Teuvo Kohonen. "Generalizations of the self-organizing map". In: *Neural Networks, 1993. IJCNN'93-Nagoya. Proceedings of 1993 International Joint Conference on*. Vol. 1. IEEE. 1993, pp. 457–462.

[132] Teuvo Kohonen. *Self-Organizing Maps*. Springer, 1995. DOI: 10.1007/978-3-642-56927-2.

[133] Barbara Hammer et al. "A general framework for unsupervised processing of structured data". In: *Neurocomputing* 57 (2004), pp. 3–35.

[134] B-H Juang and Lawrence R Rabiner. "A probabilistic distance measure for hidden Markov models". In: *AT&T technical journal* 64.2 (1985), pp. 391–408.

[135] *Classification: ROC Curve and AUC*. https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc. Accessed in May 2019.

[136] Timo Koski. *Hidden Markov models for bioinformatics*. Vol. 2. Springer Science & Business Media, 2001. ISBN: 978-1-4020-0135-2.