Simulation

Simulation: Transactions of the Society for Modeling and Simulation International
2020, Vol. 96(12) 939–956
© The Author(s) 2020
DOI: 10.1177/0037549720958480
journals.sagepub.com/home/sim
SAGE

802.11 wireless simulation and anomaly detection using HMM and UBM

Anisa Allahdadi, Ricardo Morla and Jaime S Cardoso

Abstract

Despite the growing popularity of 802.11 wireless networks, 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 in the thick of major challenges that network managers encounter. The difficulty of monitoring broad and complex Wireless Local Area Networks, that often requires heavy instrumentation of the user devices, makes anomaly detection analysis even harder. In this paper we exploit 802.11 access point usage data and propose an anomaly detection technique based on Hidden Markov Model (HMM) and Universal Background Model (UBM) on data that is inexpensive to obtain. We then generate a number of network anomalous scenarios in OMNeT ++ /INET network simulator and compare the detection outcomes with those in baseline approaches—RawData and Principal Component Analysis. The experimental results show the superiority of HMM and HMM-UBM models in detection precision and sensitivity.

Keywords

802.11 wireless network, network management, anomaly detection, Hidden Markov Model, Universal Background Model, network simulation, OMNeT ++ /INET

1. Introduction

In recent years, IEEE 802.11 wireless networks have emerged as a promising technology for wireless access by mobile devices in many public places, from enterprises and universities to urban areas. The flourishing popularity and ease of access to these networks has led to their heavy utilization and congestion. In addition, interference caused by the broadcast nature of wireless links along with other radio waves in the same frequency normally result in poor performance. In such conditions the packet transmission fails or requires several re-transmission attempts, causing performance issues. Furthermore, dynamic traffic loads, the evolving nature of user movement, and association to different access points (APs) often induce connectivity problems in large-scale 802.11 deployments. Generally speaking, at any given moment 802.11 APs or users are likely to come across problems threatening the connection quality. Thus the question of performance becomes increasingly important as new applications demand sufficient bandwidth and reliable medium access.

Across the infrastructure, there are various types of anomalous situations caused by users or APs, and automatic detection of these anomalies is of great importance for future mitigation plans. Highly utilized medium, overloaded APs, failed or crashed APs, persistent interference between adjacent APs, radio frequency (RF) effects, and authentication failure are examples of such anomalies. However, due to the time and cost limitations of constantly monitoring the entire wireless territory by sensors and sniffers,[1,2] obtaining reliable ground truth becomes more and more challenging.

In such circumstances, when acquiring ground truth is too expensive and time-consuming, network simulations seem to be effective solutions to achieve a close-to-reality setup that is computationally tractable. In the research community, many wireless networks are evaluated using discrete event simulators like OMNeT ++ .[3–5] Although having worked with other simulation frameworks such as NS3 and OPNET, we found OMNeT ++ /INET the most appropriate wireless network simulators for our research purposes. Besides the well-structured framework and user-friendly integrated development environment that facilitate

INESC TEC and Faculty of Engineering, University of Porto, Portugal

Corresponding author:
Anisa Allahdadi, INESC TEC and Campus of the Faculty of Engineering, University of Porto, Rua Dr. Roberto Frias, Porto 4200-465, Portugal.
Emails: anisa.allahdadi@inesctec.pt, anisa.allahdadi@gmail.com

analysis and data gathering, OMNeT + + /INET provides an adequate set of modules supporting physical and radio models for 802.11 that perfectly meet our requirements for this project.

In our previous papers[6,7] we utilized RADIUS authentication log data collected at the hotspot of the Faculty of Engineering of the University of Porto (FEUP). The trace data consisted of the daily summary of the connections between hundreds of APs and their corresponding wireless stations (STAs). In Allahdadi and Morla[8] we deployed a real Testbed in small scale with one AP and six STAs using FreeRADIUS server, and generated a number of anomalies in a controlled environment for experimental purposes. In the current work we simulate a more extended Wireless Local Area Network (WLAN) with five APs and 30 STAs and set up several anomalous cases, including the ones in the previous work[8] and some new anomalies. We further improve our Hidden Markov Model (HMM) formerly proposed elsewhere[7,8] by integrating it with the concept of the Universal Background Model (UBM). The simulation data are then utilized to evaluate HMM and HMM-UBM models and compare the anomaly detection results with baseline approaches (RawData and Principal Components Analysis (PCA)).

The key steps of the present work include: (a) conducting 802.11 wireless network simulation in OMNeT + + / INET to resemble normal and anomalous scenarios; (b) reiterating the simulations with different seeds to provide miscellaneous replicates; (c) extracting the wireless users' data, and converting it to AP usage data; (d) building HMM and HMM-UBM models from the prepared dataset; (e) applying the proposed anomaly detection algorithms; and (f) calculating the detection rate and sensitivity for evaluation purposes.

Regarding the anomaly detection techniques we analyze three main approaches: (a) detection of anomalous time-series in a database of time-series; (b) distinction of anomalous patterns; and (c) detection of anomalous points within a given time-series.

Furthermore, this paper explores the following research questions: (a) whether HMM and HMM-UBM models are capable of anomaly detection and anomalous pattern recognition in AP usage data; (b) whether HMM and HMM-UBM models are required for anomaly detection or the baseline approaches are enough; and (c) whether HMM-UBM have any advantages over HMM.

The rest of the paper proceeds as follows. In section 2, the related work and the most recent research relevant to the current work are presented. Section 3 briefly characterizes the data features. In section 3, the anomaly detection methodology is elaborated. Section 4 deals with the network simulation setup and focuses on the common key properties of the accomplished simulations. In section 5 the simulated scenarios are described and the experimental results are analyzed. In section 6, the main conclusions are provided and the prominent direction of future work is disclosed.

2. Related work

2.1. Anomaly detection in 802.11 wireless networks

In the most recent 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 chaotic 802.11 deployments, and similar anomalous cases.

Having explored the network under high–medium utilization conditions, Raghavendra et al.[9] showed that in the overloaded networks, stations only maintain a short association period with an AP, and repeated association and reassociation attempts are common phenomena even in the absence of client mobility. Their analysis demonstrated that stations' throughput suffers drastically from the unnecessary hand-offs, leading to sub-optimal network performance.

In another direction of work by Pan and Keshav,[10] the authors presented a number of algorithms that could detect failed APs by analyzing AP usage logs. The main assumption in their algorithm was that the longer the time an AP does not register events, the greater the probability that particular AP is faulty, crashed, or halted.

In relation to interference detection in WLANs, Broustis et al.[11] proposed methods 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, Gummadi et al.[12] 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 Clear Channel Assessment.

In Massa and Morla,[13] a usage pattern called "abrupt ending" is explored in a FEUP dataset[6,7] that concerns the disassociation of a large number of wireless sessions in the same AP within a 1 second window. The authors introduced some anomalous patterns that might be in correlation with the occurrence of this phenomenon, for instance, AP halt/crash, AP overload, persistence interference, and intermittent connectivity. The analysis of the anomaly-related patterns performed in the present research inspired our work to re-generate similar anomalies in network simulator in addition to the real Testbed that was already done in our previous work.[8] The principal goal of the simulation and the real Testbed experiments is to evaluate the HMM anomaly detection methodologies proposed in the current work as well as our former studies.[7,8]

2.2. Wireless network simulation

There are numerous efforts in the literature that tried 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.

Hernández-Campos et al.[14] employed simulation to generate synthetic traffic and validate their proposed model of traffic workload in a campus WLAN. As another example, Chen et al.[15] proposed a framework to integrate the infrastructure mode and ad hoc mode and implemented the framework in NS2. They used simulation to show the higher performance of their proposed model compared with the traditional wireless LAN. In a work rather relevant to ours, the performance of IEEE 802.11 wireless networks is evaluated using OPNET Modeler.[16] The authors investigated the performance of a pure 802.11g network over a network that uses both 802.11g and 802.11b clients by simulating network in infrastructure mode for one AP and 12 STAs. In another simulation study conducted using OPNET Modeler, IEEE 802.11b wireless LAN in a classroom network scenario is investigated.[17] The authors designed a simulation study to estimate the number of clients that can be supported in the WLAN, as well as the user-perceived web response time as a function of network load.

In relation to OMNeT + + and its simulation models, a number of researchers worked on validating the reliability and accuracy of OMNeT + + . For example, Bredel and Bergner[18] performed a measurement study of wireless networks in a highly controlled environment to validate the IEEE 802.11g model of OMNeT + + . They used metrics like throughput, delay, and packet inter-transmission to compare the measurement results with identical simulations. They showed that the simulation results match the measurements well in most cases. Furthermore, in Malekzadeh et al.[19] the reliability of OMNeT + + is assessed for wireless Denial of Service (DoS) attacks by comparing the simulation results with the real 802.11 Testbed. In this case throughput, end-to-end delay, and packet loss ratio are considered as performance measures. The authors confirm the accuracy of the simulation results in wireless DoS domain. In an important related work by Kuntz et al.[20] the extension of the OMNeT + + Mobility Framework is presented to support probabilistic propagation models. The authors provided an implementation for the Log-Normal-Shadowing, Nakagami, Rayleigh, and Rice wave propagation models and set up a framework that allows easy integration of additional models in future. Their approach is validated performing a detailed crosscheck with the network simulator NS-2.

However, there exist few efforts in the literature that conducted simulation of WLANs in OMNeT + + regarding the performance issues and quality of service (QoS). In Qashi et al.[4] the performance of the TCP protocol for audio and video transmission is evaluated using OMNeT + + simulation. In another direction of work in Woon et al.[5] 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.

In a salient line of work in Ling et al.[21] a hand-off mechanism is introduced, namely SPCC, which captures the next potential APs for context transfer in order to reduce the re-association delay. In this approach, the AP–STA link quality information is exchanged to determine the list of next potential APs. The performance of the SPCC mechanism is implemented and evaluated using the OMNeT + + simulator equipped with the INET Framework. In another study by Le et al.[22] an efficient medium access algorithm is proposed that aims at achieving time fairness and throughput enhancement in a fully distributed manner. The authors evaluated the performance of their proposed algorithm through an extensive simulation study in OMNeT + + , and the evaluation results demonstrate that the proposed algorithm leads to nearly perfect time fairness, high throughput, and low collision overhead.

To the best of our knowledge the simulation of aforementioned anomalous patterns in WLAN infrastructure mode has never been done before.

2.3. HMM applications in network analysis

In wireless networking, HMMs are employed to address various aspects of network measurement and analysis. For example, Hierarchical and Hidden Markov based techniques are analyzed in Khayam and Radha[23] to model 802.11b MAC-to-MAC channel behavior in terms of bit error and packet loss. In Kamthe et al.[24] a multilevel approach involving HMMs and Mixtures of Multivariate Bernoullis is proposed to model the long and short timescale behavior of wireless sensor network links, that is, the binary sequence of packet receptions (1s) and losses (0s) in the link. Ghosh et al.[25] applied HMMs for spectrum sensing in cognitive radios, as 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.

In another related work, HMMs are applied for modeling and prediction of user movement in wireless networks to address QoS issues.[26] 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 numbers of APs, not widespread WLANs. Cheikh et al.[27] proposed a new approach for optimizing the hand-off decision in Femtocell networks using HMM. They applied HMM to predict the target Femtocell Access Point by observing the geographic positions of the mobile user.

Kashyap et al.[28] and Paul et al.[29] 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 HMM, and learned the state transition probabilities in this model using the observed traces. 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).

As the above literature indicates, HMM-related studies in wireless network management are rarely used specifically in performance anomaly detection of wireless networks.

3. Data features

In our previous papers we utilized RADIUS authentication log data, which contain session records of wireless stations connecting to APs. A preliminary analysis on the raw data yields a sequential dataset summarizing APs association history.[8] In the current simulation we create a similar dataset with the exact same features to be synchronized with the previous HMM modeling. The definition of the main features along with a brief explanation on the feature selection process is presented in the following paragraphs.

3.1. Data attributes

Data features are categorized in two main classes: *Density Attributes* and *Usage Attributes. Density Attributes* demonstrate how crowded is the place in terms of active attendant users, and the *Usage Attributes* disclose the volume of the sent and received traffics by the present users. The former attributes mainly characterize the association population and durability, while the latter attributes reveal the total bandwidth throughput regardless of how populous is the place, and is more relevant to the applications utilized by the current mobile users.

3.1.1. Density attributes

User count. This is the number of unique users observed in a specific location (indicated by an AP) in a time-slot.

Session count. This is the total population of active sessions during a time-slot regardless of the owner user. This attribute reveals the number of attempts made by the congregation of the present users to associate to the current AP.

Connection duration. This is the total duration of association time of all the current users. This attribute is an indicator of the overall connection persistence. The utmost amount of this feature is achieved when there is no evidence of disassociation in the ongoing active session during a time-slot.

3.1.2. Usage attributes

Input data in octets. This is the number of octets transmitted from the client. This attributes briefly refers to the number of bytes uploaded by the wireless user.

Output data in octets. This is the number of octets received by the client. This attribute shortly refers to the number of bytes downloaded by the wireless user.

Input data in packets. This is the number of packets transmitted from the client. This attribute is similar to the above *Input-Octet*, just to be measured in packets.

Output data in packets. This is the number of packets received by the client. This attribute is similar to the above *Output-Octet*, just to be measured in packets.

3.2. Feature selection

For subsequent analysis, we favor using fewer features than the entire set of attributes introduced earlier. For this purpose, we applied the PCA technique to find the combination of the variables which best explain the phenomena and contain the greatest part of the entire information. In the current experiment the first three principal components bring the cumulative proportion of variance to over 99%. More detailed explanations on the correlation of data features with themselves and with the principal components are provided in our previous work.[8]

4. Anomaly detection in AP usage data

We use HMMs adapted from a UBM for (a) detection of anomalous time-series, (b) distinction of anomalous patterns, and (c) detection of anomalies within a given time-series.

4.1. Preliminaries

4.1.1. HMM. The HMM symbolizes a doubly stochastic process with a set of observable states and a series of hidden states which can only be observed through the observable set of stochastic process. HMMs are generally used for the stochastic modeling of non-stationary time-series.

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:

$$P(O, H|\lambda) = P(h_0) \prod_{l=1}^{T} P(h_l|h_{l-1}) P(o_l|h_l), \tag{1}$$

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:

$$P(o_t|h_t) = P(o_{t'}|h_{t'}), \tag{2}$$

$$P(h_t|h_{t-1}) = P(h_{t'}|h_{t'-1}), \text{for all } t' \tag{3}$$

Thus, an HMM is completely defined by the following parameters:

- The number of hidden states, N.
- The discrete set of hidden states, $S = \{s_i\}$, $1 \leqslant i \leqslant N$.
- If observations are discrete, the number of possible observations, M, and the discrete set of such observations, $V = \{v_k\}$, $1 \leqslant k \leqslant M$.
- 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 $A = [a_{i,j}]$, $1 \leqslant i,j \leqslant N$, 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 $B = [b_i(v_k)]$, $1 \leqslant i \leqslant N$, $1 \leqslant k \leqslant M$, 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, $\Sigma = \{\Sigma_i\}$, $1 \leqslant i \leqslant N$.
- The initial state probability distribution, $P(h_0)$, represented by a vector $\pi = [\pi_i]$, $1 \leqslant i \leqslant N$, where $\pi_i = P(h_0 = s_i)$.

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.

4.1.2. UBM.

A 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.[30]

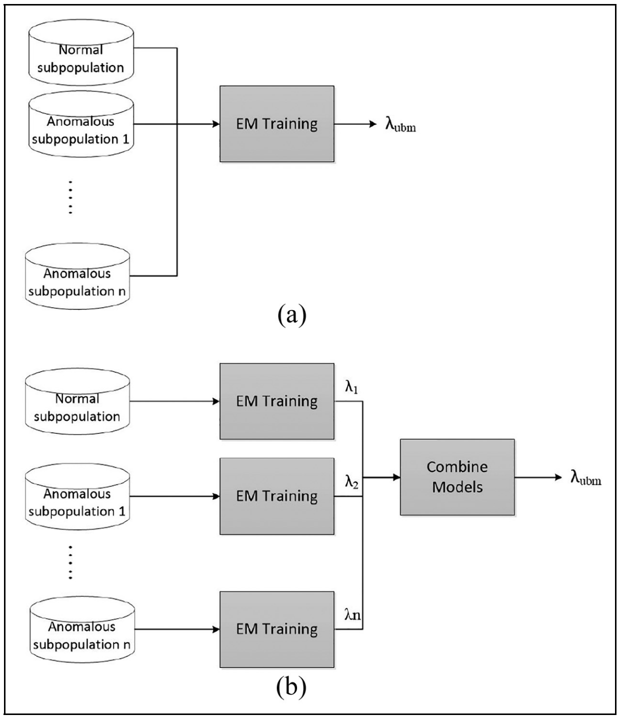

Figure 1. Data and model pooling approaches for creating a UBM. (a) Data from sub-populations pooled prior to training the final UBM. (b) Individual sub-population models trained then combined to create final UBM.

We applied UBM to initialize the HMM models using the data available from all AP experiments regardless of whether they contained anomalies or not. This is advantageous, as in the 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 6.

Given the data to train 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 1(a)). One should be careful that the pooled data are balanced over the sub-populations within the data, otherwise the final model will be biased toward the dominant sub-population.[31] Another approach is to train individual UBMs over the sub-populations in the data, and then pool the sub-population models together (Figure 1(b)). The latter approach has the advantages that one can effectively use unbalanced data and can carefully control the composition of the final UBM.[31] 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 dataset contains normal samples and the rest consist of anomalous events (equal portion for each anomaly).

4.2. Anomaly detection

4.2.1. 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 which contain a number of anomalous events. Similar to traditional outlier detection methods (we use "outlier" and "anomaly" interchangeably in this context), the usual approach is to train 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.[31] In our case, we build an HMM model with UBM initialization using the training data of all the experiments. Then we calculate the log-likelihood values of each time-series in the test dataset. Those experiments that contain one or more anomalous events are expected to gain lower log-likelihood values.

The likelihood value of HMM is the probability of an observation sequence given the model parameters. Equation 4 shows how the likelihood value of HMM model λ is calculated.

$$\begin{aligned} P(O|\lambda) &= \sum_{all\,S} P(O|S,\lambda)P(S|\lambda) \\ &= \sum_{s_1,s_2,\ldots s_T} \pi_{s_1} b_{s_1}(O_1) a_{s_1,s_2} b_{s_2}(O_2) \ldots a_{s_{T-1},s_T} b_{s_T}(O_T) \end{aligned} \tag{4}$$

Due to the vanishingly small likelihood probabilities produced in long time-series, normally the logarithmic value is utilized.

Figure 2 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, whereas the values of the AP Shutdown/Halt scenario are condensed in a limited range.

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.

4.2.2. Distinction of anomalous patterns. To capture distinctive characteristics of the anomalous scenarios we build separate HMM models for each one and also one model for the normal scenario. Then we compute the probability of each observation sequence getting generated by each of these models. The HMM model that produces the highest log-likelihood value is considered to be the generative model of the given time-series.

Choosing the best λ model among the competing models is termed as the *scoring problem* and is a function of log-likelihood values. At the end of this process we obtain a 2D matrix whose rows and columns consist of HMM models and observation sequences, respectively.

Figure 3 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 bottom part of the bars (in blue) demonstrate the percentage of time-series correctly detected by their corresponding models. The top piece of the bars (in pink) shows the mis-detection ratio that occurs in AP Overload and Flash Crowd scenarios. Some 25% of AP Overload time-series 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 work contains a single anomaly, whereas in reality each time-series can contain no anomaly (in normal cases) or various 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 mode. Here we propose the basic scheme of an algorithm which is based on the general model training in Zhang et al.,[32] 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).

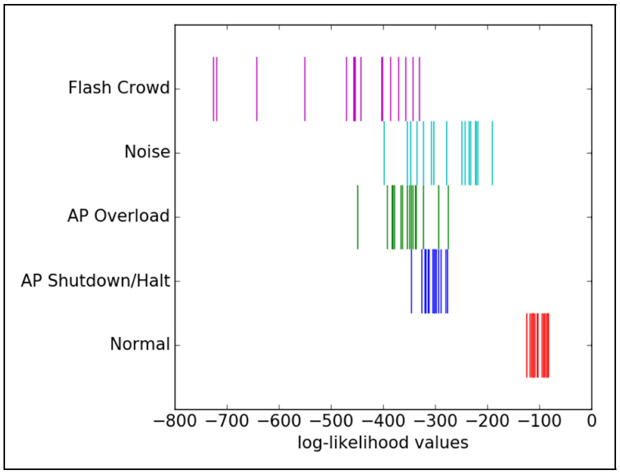

Figure 2. Log-likelihood values of normal and anomalous experiments.

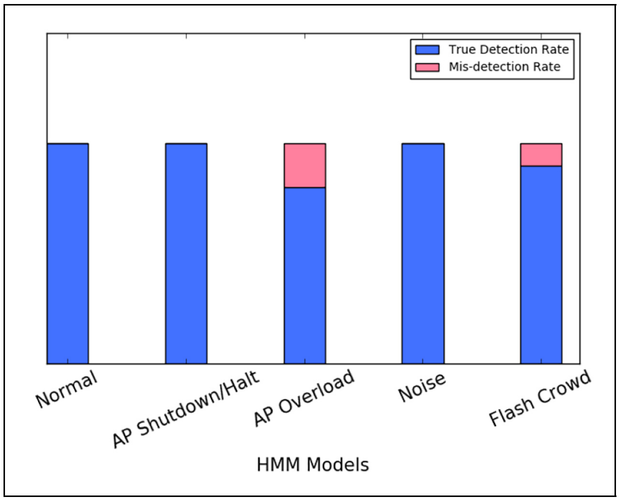

Figure 3. Detection results of the observations data by the trained HMM models.

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 belongs to any previous model).
6. Repeat step 4 and 5 until there are no more test sequences.

There are a number of parameters in this algorithm that are 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 paper and is left for the future work.

4.2.3. 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 accordingly 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 also produces false positives; however, in all the performed experiments of this work the false positive ratio is insignificant.

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 $s_1 s_2 \ldots s_{40}$, the algorithm detects s_2

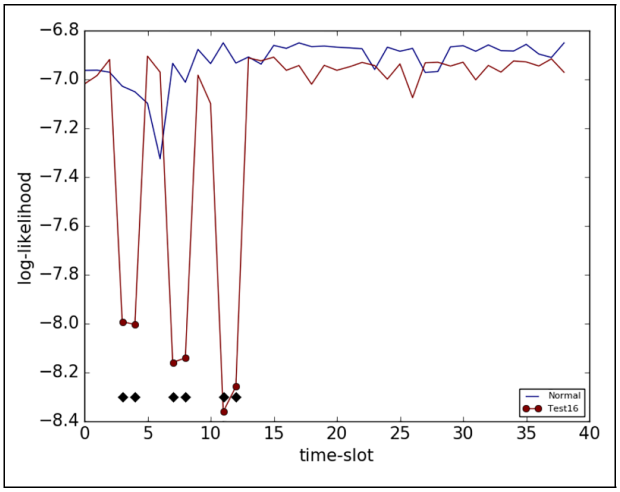

Figure 4. Log-likelihood of the normal model together with an example anomaly related to AP Overload experiment.

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

We further explore this type of anomaly detection in Section 6 and analyze each anomalous case specifically in more detail.

5. Experimental setup

To evaluate the proposed strategy, we perform an extensive set of simulations using OMNeT ++ [33] simulator and INET framework.[34] 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.

In a DES, as well as the OMNeT ++ , events take place at discrete instances in time, and they take 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 it based on the number of nodes, amount of traffic transferred, and other details of the network. In our example, with the current number of nodes (five APs and 30 STAs) and traffic plan, 10 minutes of simulation time takes around 17 minutes of CPU time. Our HMM approach operates on 40 consecutive time-slots of 15 s simulation time each.

5.1. Normal scenario

Figure 5(a) shows the initial picture of a normal scenario, the location of the APs, STAs, and the servers. Figure 5(b) displays the location of the STAs after passing 30 s (simulation time) from the beginning of the simulation.

In the normal scenario, there are five APs and 30 STAs. Each STA is initially associated to one of the available APs depending on its location. During the simulation STAs, based on their mobility models, are handed over to other APs when moving around the simulation ground. Furthermore, according to the defined traffic plans in section 5.3, each node sends and receives packets to the existing servers.

5.2. 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 the first AP (AP1) follow the *Linear Mobility* pattern which is configured with speed, angle, and acceleration parameters. 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 the other AP (AP2) besides their own AP (AP1), and sometimes 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.

The nodes connected to the second AP (AP2) follow the *Mass Mobility* model, and accordingly move within the room. This pattern of mobility is intended to represent places like a classroom or library in which users do not leave the place frequently, but still have some motions in the place.

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 the node reaches the target position, it waits for a specified waitTime and selects a new random position afterward. This type of movement resembles the

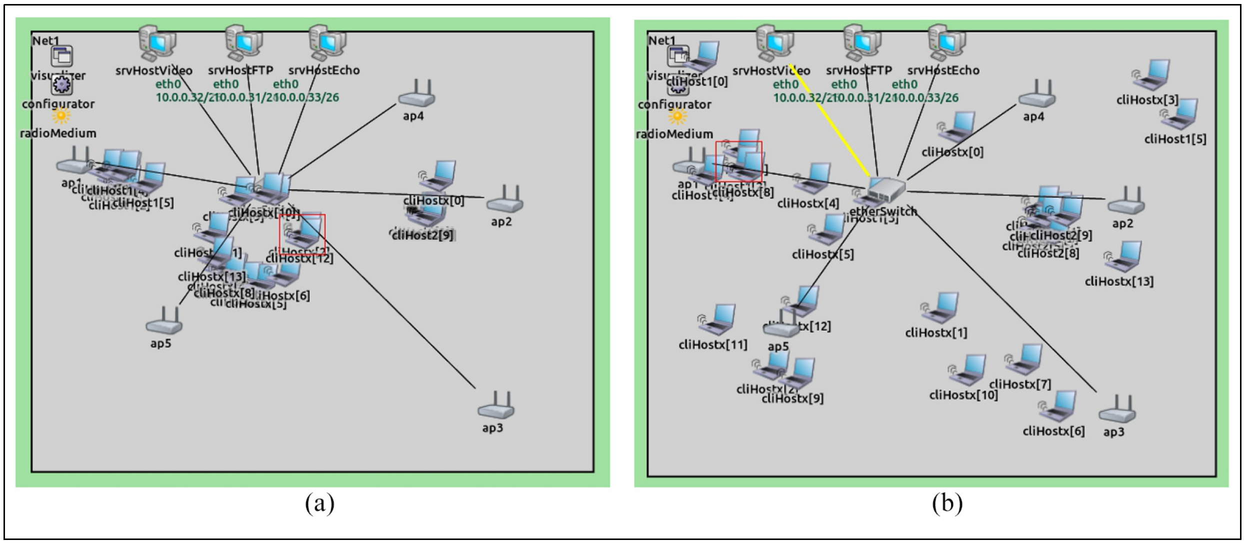

Figure 5. (a) The initial picture of the wireless network simulated in OMNeT++/INET. (b) Location of the wireless stations after 30 s of simulation.

Table 1. Wireless nodes' specifications in terms of mobility models.

Mobility Model	# Nodes	Mobility Parameters
Linear Mobility	6	speed: truncnormal (20mps, 10mps)
Mass Mobility	10	speed: truncnormal (70mps, 50mps)
Random Waypoint Mobility	14	speed: uniform (50mps,50mps)

random mobile users around the wireless ground mostly connected with their mobile devices.

A summary of wireless nodes' specifications in terms of mobility models is provided in Table 1.

5.3. Traffic generation

As shown in Figure 5(a) and 5(b), 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 the clients of AP2, resembling the video downloading by those users. The FTP server (srvHostFTP) is to receive the FTP uploads by the clients of AP1 with message length of $\mathcal{N}(500B, 100B)$. In addition to exclusively downloading or uploading, the other server (srvHostEcho) is in charge of sending and receiving traffic to all the users. This traffic pattern represents the common act of email checking and web browsing by the wireless nodes. The echo packets length are configured to be smaller than the previous ones, $\mathcal{N}(200B, 50B)$, indicating a lighter traffic transmission. In the *AP Overload* anomalous scenario one more server is added to take care of heavy channel utilization (srvHostBurst), and more detail about that can be found in section 6.2.

5.4. 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.

6. Experimental results and evaluation

In this section we explore a set of anomalous scenarios and describe different cases of each one. Then we present the HMM and HMM-UBM results in anomaly detection and compare them with baseline approaches (RawData and PCA) for evaluation purposes.

In terms of HMMs, we consider fully connected models (ergodic), continuous observations with Gaussian distributions, and three hidden states. We believe that HMMs with two 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 four or a higher number of states. Each experiment is repeated at least 20 times with different seeds in order to examine the models on miscellaneous samples providing slightly different data. Eighty percent of the data sequences is used for training the model and 20% is kept for testing.

6.1. 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 6 demonstrates the HMM likelihood series and the anomalies detected for the test dataset 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 even short shutdown periods that only last for one time-slot. However, Figure 6(a) shows that the HMM model built with only normal data gives a clearer model rather than the HMM-UBM model built with the entire dataset including the anomalous experiments (Figure 6(b)). Despite this, HMM-UBM obtains adequate values for precision and recall, and even higher precision results in some cases.

Figure 7 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 a higher precision value and smaller false positive ratio compared with 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 dataset features when the power is gone and no session is recorded. That is the reason RawData attains 100% recall. However, it produces a relatively high false positive result that yields low precision.

6.2. AP overload

In this anomalous case, excessive channel utilization occurs that could be the consequence of excessive

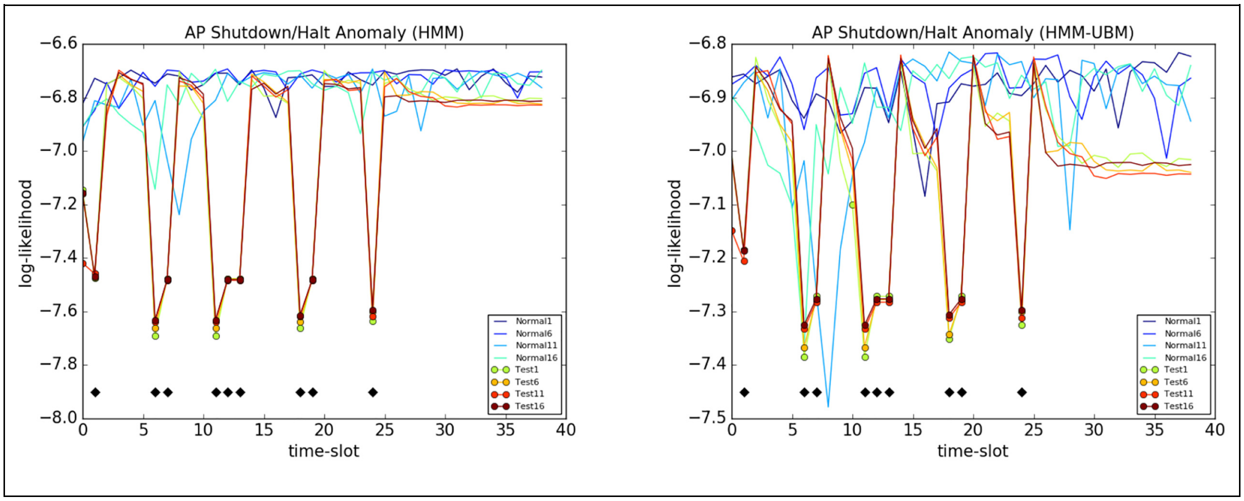

Figure 6. The log-likelihood series and detected anomalies of AP shutdown/halt scenario in HMM and HMM-UBM models. (a) HMM; (b) HMM-UBM.

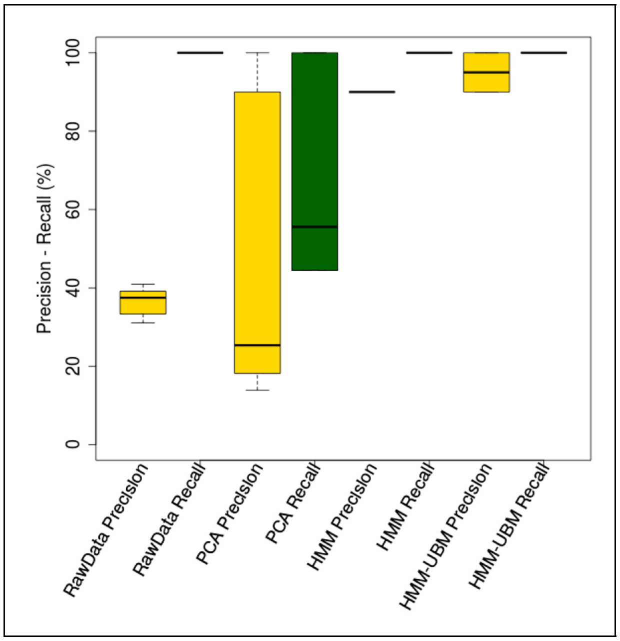

Figure 7. Precision and recall boxplot of RawData, PCA, HMM, and HMM-UBM belong to AP shutdown/halt scenario.

download or upload by a number of wireless users. In such circumstances, the clients could 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 the second AP. Burst server (srvHostBurst) sends UDP packets to the given IP addresses in bursts during the *burst-duration* which resembles the heavy downloads of the wireless users. In the time of *sleep-duration* the burst flow stops and the channel utilization gets back to normal. This experiment contains three different cases as follows:

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

Figure 8 and 9 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, respectively. As shown in these figures, during the burst period the log-likelihood value drops 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 10 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 detect directly from the raw data and needs some advanced techniques. The HMM and HMM-UBM results, both in precision and recall, outperform the baseline approaches.

6.3. 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 does not come from a particular source, nor propagate 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 the *IsotropicBackground Noise* parameter in the simulator. The default value of this parameter is set to –110 dBm which is the minimum noise level in Wi-Fi networks 802.11 variants. We gradually increase the noise power to –90 dBm and record the simulation results repeated 10 times for each experiment. According to Fu et al.,[35] the average noise level in a busy university campus had a stable value at around –94 dBm.

Figures 11 and 12 demonstrate the log-likelihood series of this anomalous scenario, and like previous cases the

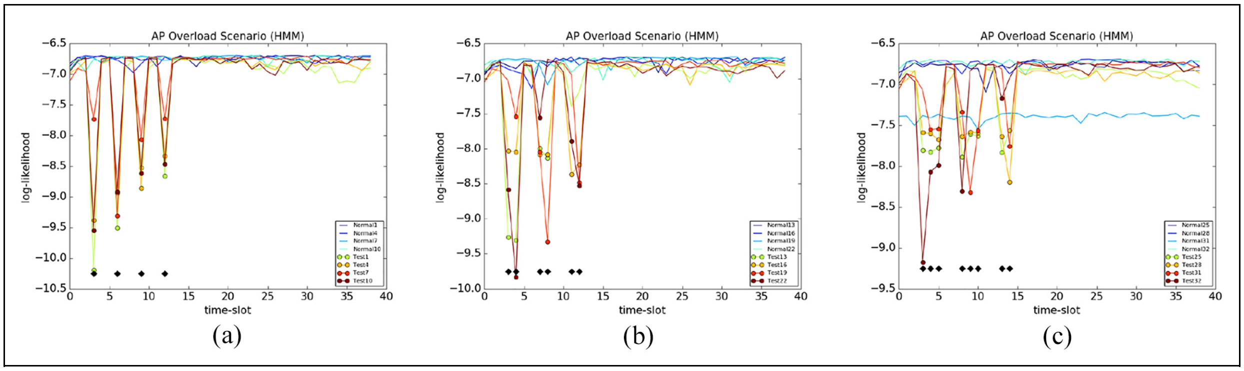

Figure 8. The log-likelihood series and detected anomalies of AP overload scenario (HMM).
(a) burst-duration $<$ sleep-duration; (b) burst-duration $=$ sleep-duration; (c) burst-duration $>$ sleep-duration.

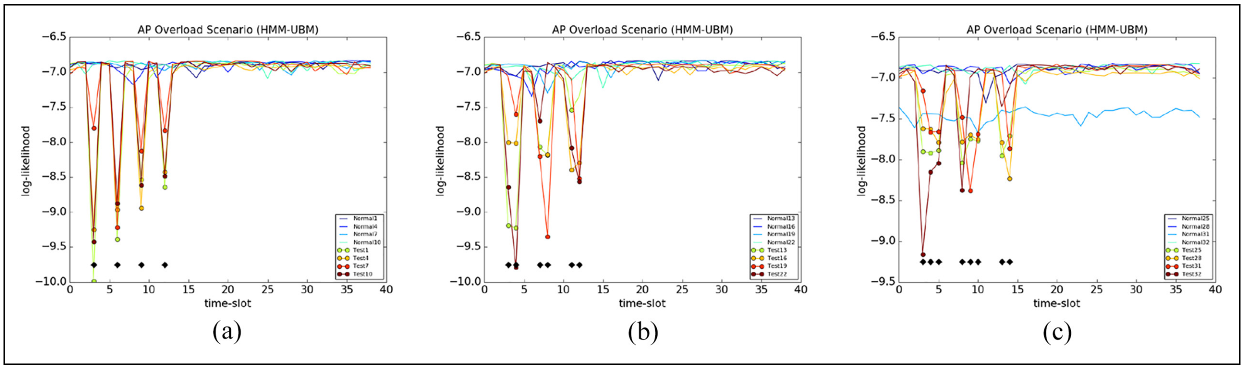

Figure 9. The log-likelihood series and detected anomalies of AP overload scenario (HMM-UBM). (a) burst-duration < sleep-duration; (b) burst-duration = sleep-duration; (c) burst-duration > sleep-duration.

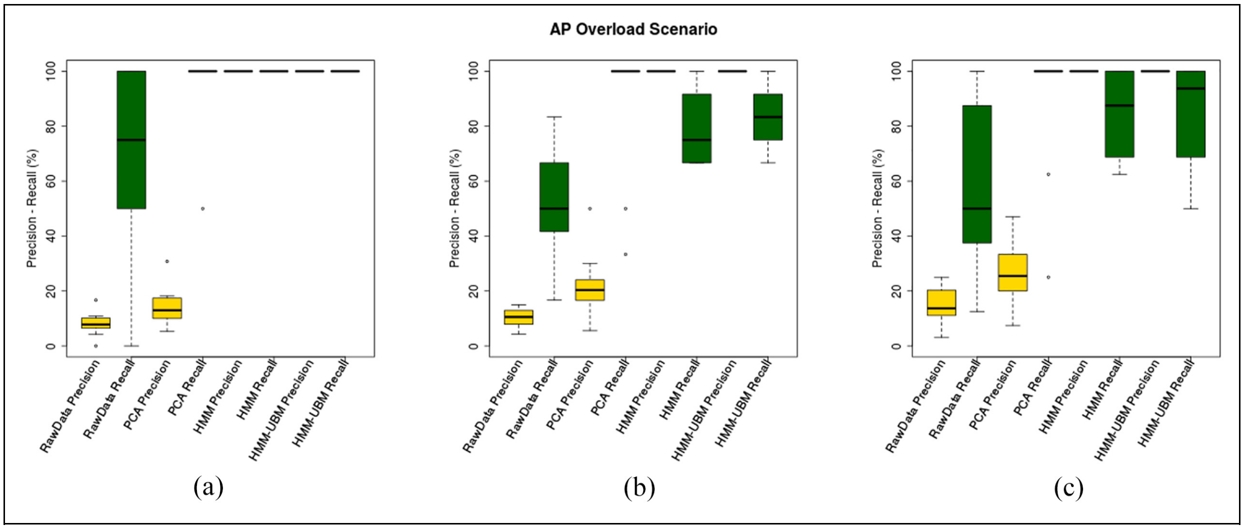

Figure 10. Precision and recall boxplot of RawData, PCA and HMM belong to AP overload scenario. Left: burst-duration < sleep-duration; middle: burst-duration = sleep-duration; right: burst-duration > sleep-duration.

valley shapes represent the anomalies. The simulated anomalous period is during the first 10 time-slots, and 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 become closer to the normal case (noise power of −110 dBm).

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 13 shows, HMM and HMM-UBM models in all the experiments present higher precision values than RawData and PCA.

6.4. Flash crowd

In wireless networks an unexpected surge of traffic can occur at 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 HMM and HMM-UBM model is able to detect such alterations in the normal usage pattern, we simulate this example in two experiments:

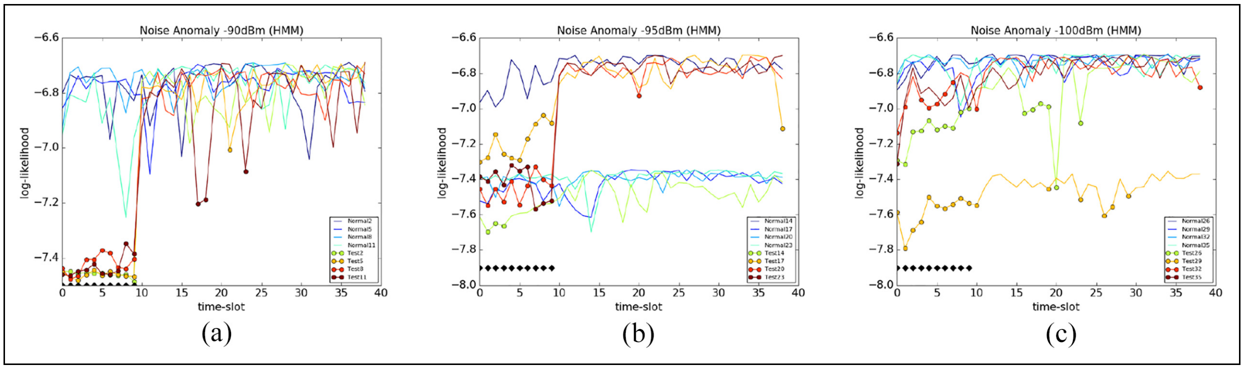

Figure 11. The log-likelihood series and detected anomalies of Noise scenario (HMM). –90 dBm; (b) –95 dBm; (c) –100 dBm.

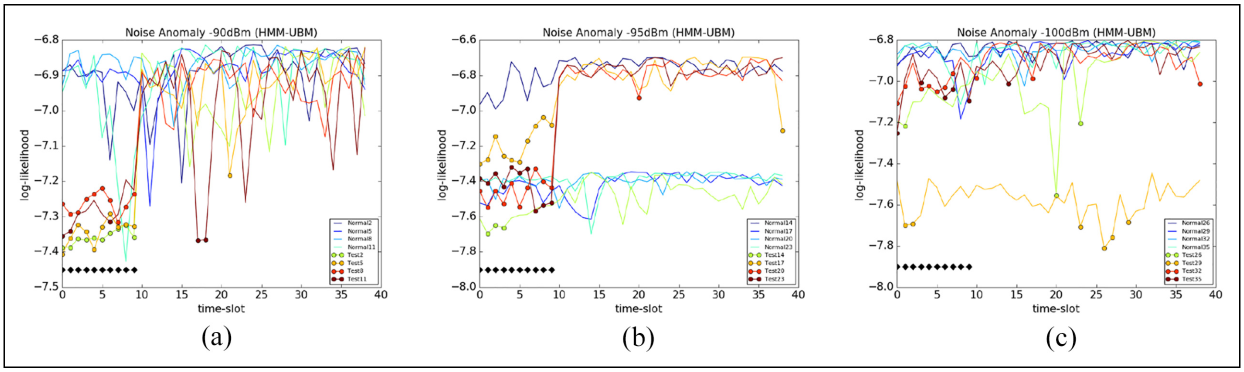

Figure 12. The log-likelihood series and detected anomalies of Noise scenario (HMM-UBM). –90 dBm; (b) –95 dBm; (c) –100 dBm.

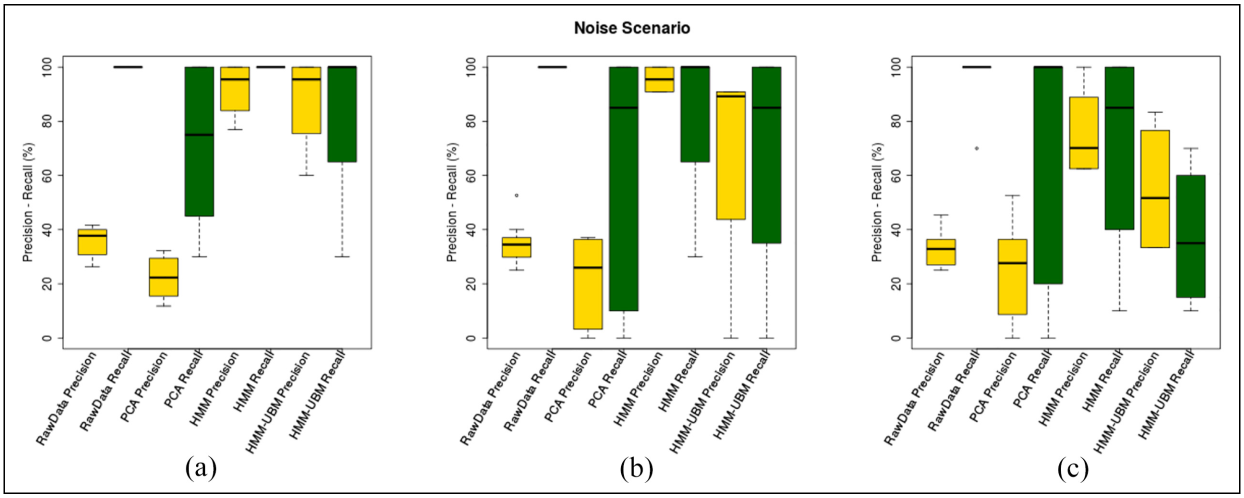

Figure 13. Precision and recall boxplot of RawData, PCA, HMM, and HMM-UBM belonging to the noise scenario. Left: –90 dBm; middle: –95 dBm; right: –100dBm.

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

Figures 14 and 15 present the log-likelihood series of Flash Crowd scenario, and detected anomalous points as colored circles and simulated anomalies as black diamonds. In only one test case in the departure scenario, which is related to

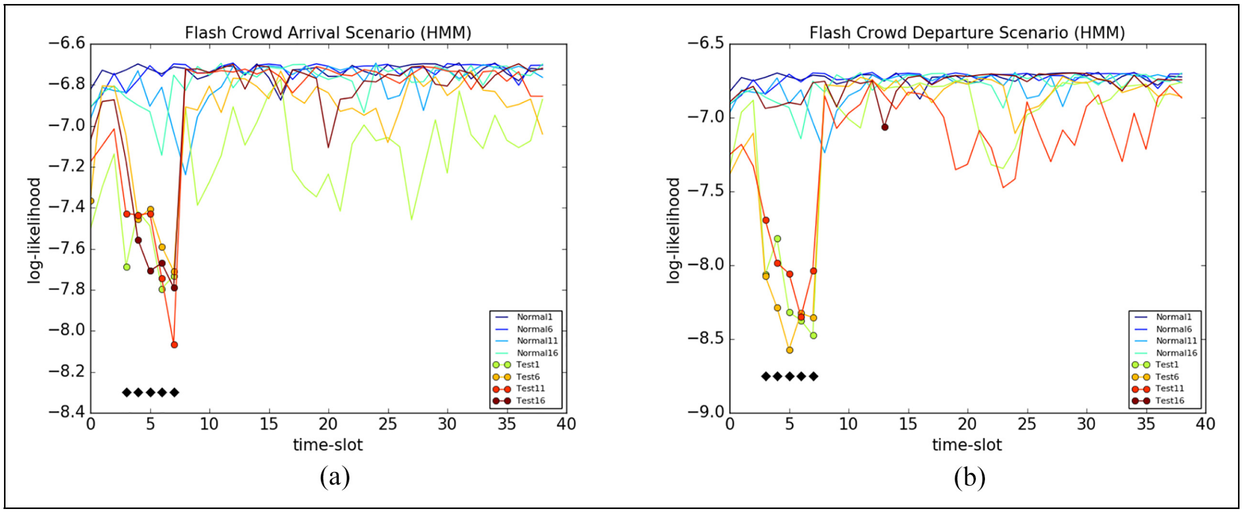

Figure 14. The log-likelihood series and detected anomalies of Flash Crowd scenario (HMM). (a) Flash Crowd Arrival Scenario; (b) Flash Crowd Departure Scenario.

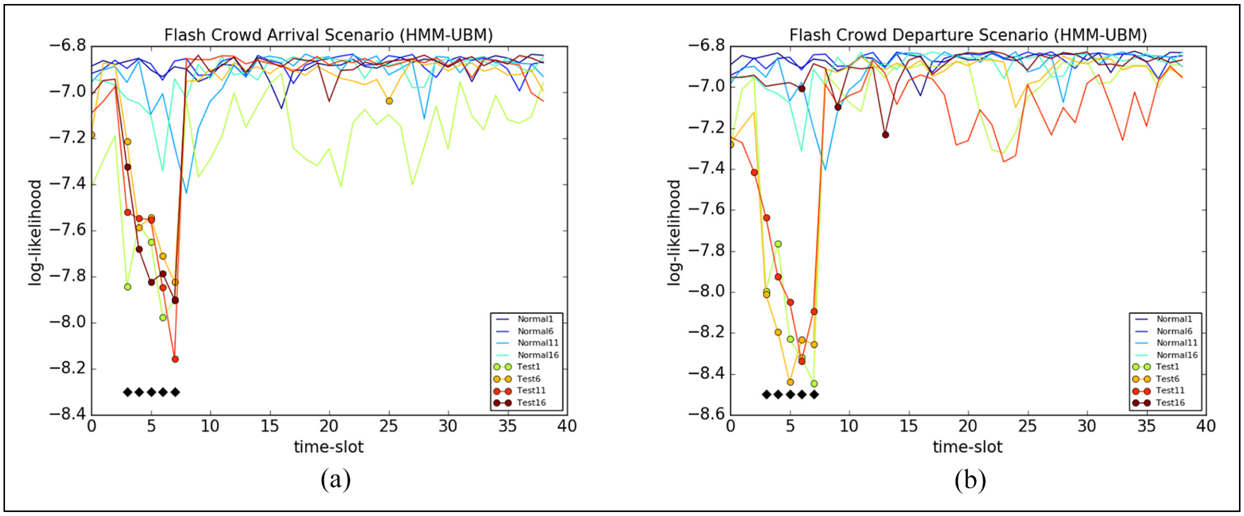

Figure 15. The log-likelihood series and detected anomalies of Flash Crowd scenario (HMM-UBM). (a) Flash Crowd Arrival Scenario; (b) Flash Crowd Departure Scenario.

Rician Fading path loss, is the anomalous period not detected, either in HMM or in HMM-UBM. In the rest of the experiments the anomaly detection technique performs accurately both in arrival and departure scenarios.

As illustrated in the boxplot diagram of Figure 16, 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.

7. Simulation validation

The main objective of the simulation, conducted in OMNeT ++ -5.4.1 and INET-4.0.0, is to achieve synthesized data for evaluation of the proposed methodologies. Having used the existing library of components from the INET framework, we were able to put together the required submodules to set up our IEEE 802.11 Wireless network. The design was made using the GNED tool of the OMNeT ++ . Four anomalous scenarios are simulated along with a normal scenario. We determined the main structure of the simulation model in one NED file while each anomalous scenario contains a separate NED file that inherits from the main NED. In each anomalous scenario the normal process of the simulation is interrupted by changing hyper parameters of the model (e.g., in omnetpp.ini) or by running scripts (via ScenarioManager) to provoke the desired anomalous effects in specified times. Table 2 shows the list of principal INET

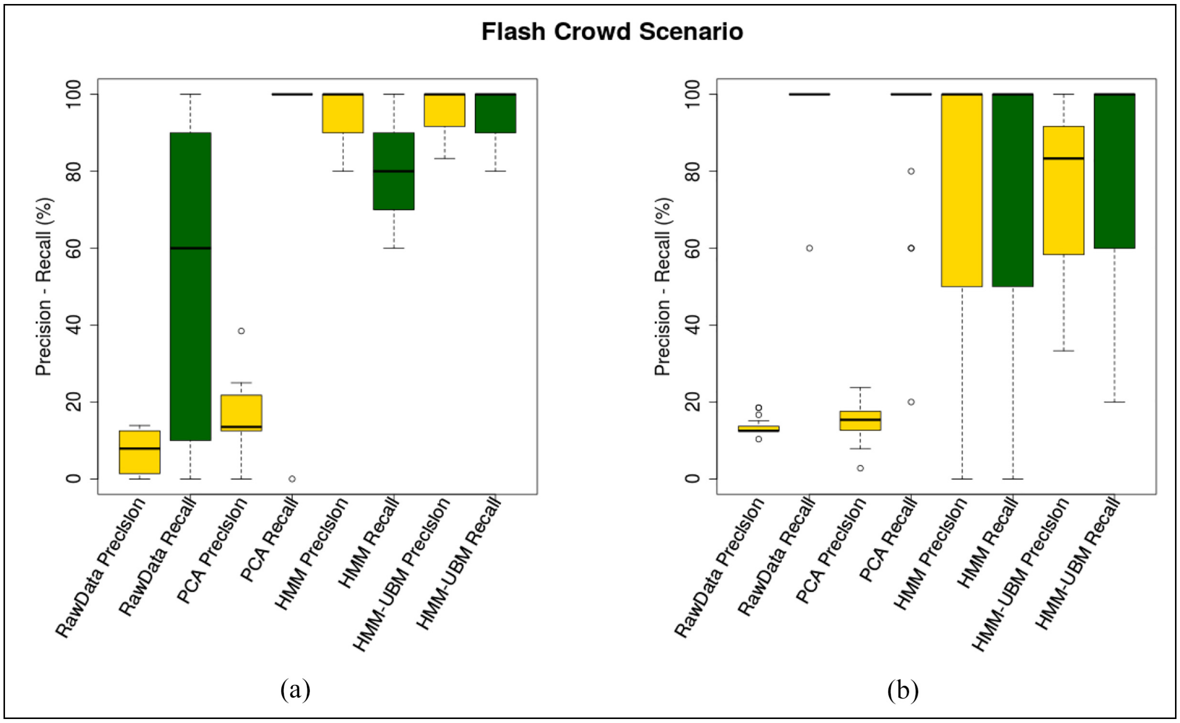

Figure 16. Precision and recall boxplot of RawData, PCA, and HMM belong to flash crowd scenario. Left: arrival scenario; right: departure scenario.

Table 2. INET submodules used for the simulation of all experiments.

INET submodule	Description
inet.node.inet.AccessPoint	AP[1..5].
inet.node.inet.WirelessHost	STA[1..30].
inet.node.inet.StandardHost	Servers[1..4].
inet.node.inet.Eth10G	connecting AP[1..5].ethg++ to EtherSwitch.ethg++.
inet.node.ethernet.EtherSwitch	as a connection between APs and servers.
inet.physicallayer.ieee80211.packetlevel. Ieee80211ScalarRadioMedium	the shared physical medium in charge of radios, noise sources, and ongoing transmission.
inet.networklayer.configurator.ipv4. Ipv4NetworkConfigurator	for assigning IP addresses to network nodes and setting up their routing tables.
inet.common.scenario.ScenarioManager	to schedule certain events to take place at specified times, used in AP Shutdown/ Halt, Noise, and Flash Crowd scenarios.
inet.examples.httptools.socket. tenserversocket.ethernetline	connecting Servers[1..4].ethg++ to EtherSwitch.ethg++.
inet.visualizer.integrated.IntegratedCanvasVisualizer	support the visualization of the nodes in runtime.

submodules used for the simulation of the normal and anomalous experiments. A number of these components are displayed in Figure 5.

For the implementation of the aforementioned scenarios we made use of several projects from INET examples as starting points; for instance shutdownrestart, udpburst, udpclientserver, handover, hiddennode, lan80211, qos, and wiredandwirelesshostwithap, among others. The entire list of INET examples can be found in the INET framework documentation.[36] Upon the accomplishment of each set of experiments, we accurately inspected the final results using the output vectors extracted from *anf* files. The *anf* files provide the graphical analysis of output data and facilitate data gathering from all the wireless nodes (APs, STAs, Servers). Succeeding each set of experiment, the wireless users' information was extracted and stored separately in a systematic approach for further analysis. This information contains hand-off data of the STAs moving around the wireless ground associating to nearby APs, the amount of traffic transferred between STAs and APs, and the association time of STAs to the available APs. In a posterior analysis implemented in R,[37] the STA usage data were converted to AP usage data. To do so, we aggregated the aforementioned information from all the connected STAs to a given AP and computed the density and usage attributes of that AP as described in Section 3.1. For instance, the number of STAs connected to an AP, or the total connection time to an AP, or the amount of traffic transferred to/from an AP in a given time-slot determine the AP usage characteristics in a timely manner.

Having analyzed the AP usage dataset, we cross-checked the validity of anomalous scenarios as follows:

- AP Shutdown/ Halt: This anomaly is reproduced by turning off the AP power during the *halt-period* for a number of *time-slots*. As a result, in AP usage data there must not be any record of STAs connecting to that AP during the *halt-period*.
- AP Overload: In this case AP heavy usage is simulated through a burst server sending UDP packets to the given IP addresses in bursts during the *burst-duration*. In the time of *sleep-duration* the burst flow stops and the channel utilization gets back to normal. These fluctuations between burst and normal utilization are reflected in the usage attributes of the anomalous AP in AP usage data.
- Noise: This scenario is simulated by changing the level of noise power gradually via adjusting the value of *IsotropicBackgroundNoise* parameter. *OutputOctets* and *OutputPackets* features from the usage attributes demonstrate the amount of modification in noise power.
- Flash Crowd: This scenario focuses on simultaneous association/disassociation of a number of wireless users to/from an AP. In arrival and departure scenarios both density and usage attributes encounter a sudden increase and decrease, respectively.

8. Simulation application in real network

The network we simulated in this work is a reproduction of the Eduroam European wireless academic network on a much smaller scale. Our initial intention was to propose HMM-related methodologies to characterize the usage pattern of APs in university hotspots to provide models for anomaly detection. For this purpose, we utilized the log data of RADIUS authentication collected from FEUP. This large dataset, however, does not contain any ground truth of anomalous events. To evaluate our proposed anomaly detection techniques, we deployed a small Testbed and generated a few numbers of anomalous cases in a controlled environment in our previous paper.[8] The deployed Testbed include one AP and six wireless stations, and the provoked anomalies addressed issues regarding AP Shutdown/ Halt, Heavy Usage, and Interference. We decided to conduct wireless simulation to extend the set of deployed anomalies and to enlarge the wireless network including the users' mobility between neighboring APs. The structure of the wireless network and the schema of the data collected from the real network at FEUP, Testbed, and simulation are identical and related to the RADIUS authentication data collected at APs. Analysis of a large dataset obtained from a real network in a university campus has numerous advantages; however, in anomaly detection domain acquisition of ground truth is extremely essential. The Testbed deployment and wireless simulation provided us with a labeled dataset similar to the original RADIUS data though on a smaller scale. In the current paper, as well as our previous one,[8] we showed how the modeling and anomaly detection techniques operate in the presence of the ground truth.

9. Conclusions and future work

Intelligent detection of anomalies in 802.11 networks from the analysis of the collected AP usage data is of great significance to network managers. It facilitates their everyday administration workload, and assists them in network maintenance, providing future mitigation plans.

The key contributions of this work consist of: (a) HMM modeling and threshold detection technique for anomaly detection; (b) proposing HMM-UBM technique for a robust initialization of the hidden states and unsupervised learning; and (c) 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 with 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 small false positive ratio. This is promising, for in the 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 for anomaly detection purposes.

In future work we intend to propose a hybrid HMM model that considers the spatial proximity of APs in addition to the temporal relativity of data sequences. Furthermore, we intend to propose an unsupervised learning algorithm for modeling and characterizing various anomaly-related patterns.

Funding

This work is financed by the ERDF—European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation—COMPETE 2020 Programme within project POCI-01-0145-FEDER-006961, and by National Funds through the FCT—Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) as part of project UID/EEA/50014/2019. The first author is also sponsored by FCT grant SFRH/BD/99714/2014.

ORCID iD

Anisa Allahdadi https://orcid.org/0000-0002-0834-4049

References

1. Cheng YC, Bellardo J, Benkö P, et al. Jigsaw: Solving the puzzle of enterprise 802.11 analysis. In: *Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*. SIGCOMM '06, New York, NY, USA: ACM. ISBN 1-59593-308-5, pp.39–50.
2. Sheth A, Doerr C, Grunwald D, 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, pp.191–204.
3. Colesanti UM, Crociani C and Vitaletti A. On the accuracy of omnet++ in the wireless sensornetworks domain: simulation vs. testbed. In: *Proceedings of the 4th ACM workshop on performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*. ACM, pp.25–31.
4. Qashi R, Bogdan M and Haenssgen K. 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, p.M48.
5. Woon S, Wu E and Sekercioglu A. A simulation model of IEEE802. 11b for performance analysis of wireless LAN protocols. In: *Australian Telecommunications, Networks and Applications Conference (ATNAC)*, volume 162.
6. Allahdadi A, Morla R, Aguiar A, et al. Predicting short 802.11 sessions from radius usage data. In: *2013 IEEE 38th Conference on Local Computer Networks Workshops (LCN Workshops)*. IEEE, pp.1–8.
7. Allahdadi A, Morla R and Cardoso JS. Outlier detection in 802.11 wireless access points using hidden Markov models. In: *2014 7th IFIP Wireless and Mobile Networking Conference (WMNC)*. IEEE, pp.1–8.
8. Allahdadi A and Morla R. Anomaly detection and modeling in 802.11 wireless networks. *J Netw Syst Manag* 2019; 27(1): 3–38.
9. Raghavendra R, Belding EM, Papagiannaki K, et al. Unwanted link layer traffic in large IEEE 802.11 wireless networks. *IEEE Trans Mobile Comput* 2010; 9(9): 1212–1225.
10. Pan H and Keshav S. Detection and repair of faulty access points. In: *Wireless Communications and Networking Conference, 2006. WCNC 2006. IEEE*, volume 1. IEEE, pp.532–538.
11. Broustis I, Papagiannaki K, Krishnamurthy SV, et al. Measurement-driven guidelines for 802.11 WLAN design. *IEEE/ACM Trans Netw (TON)* 2010; 18(3): 722–735.
12. Gummadi R, Wetherall D, Greenstein B, et al. Understanding and mitigating the impact of RF interference on 802.11 networks. *ACM SIGCOMM Comput Commun Rev* 2007; 37(4): 385–396.
13. Massa D and Morla R. Abrupt ending of 802.11 AP connections. In: *2013 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, pp.000348–000353.
14. Hernández-Campos F, Karaliopoulos M, Papadopouli M, et al. Spatio-temporal modeling of traffic workload in a campus WLAN. In: *Proceedings of the 2nd annual international workshop on wireless internet*. ACM, p.1.
15. Chen J, Chan SH and Liew SC. Mixed-mode WLAN: The integration of ad hoc mode with wireless LAN infrastructure. In: *Global telecommunications conference, 2003. GLOBECOM'03. IEEE*, volume 1. IEEE, pp.231–235.
16. Kulgachev V and Jasani H. 802.11 networks performance evaluation using OpNet. In: *Proceedings of the 2010 ACM conference on information technology education*. ACM, pp.149–152.
17. Bai G and Williamson C. Simulation evaluation of wireless web performance in an IEEE 802.11 b classroom area network. In: *28th annual IEEE international conference on local computer networks, 2003. LCN'03. Proceedings*. IEEE, pp.663–672.
18. Bredel M and Bergner M. 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), p.81.
19. Malekzadeh M, Ghani AAA, Subramaniam S, et al. Validating reliability of OmNet++ in wireless networks DOS attacks: Simulation vs. testbed. *IJ Netw Secu* 2011; 13(1): 13–21.
20. Kuntz A, Schmidt-Eisenlohr F, Graute O, et al. Introducing probabilistic radio propagation models in omnet++ mobility framework and cross validation check with ns-2. In: *Proceedings of the 1st international conference on simulation tools and techniques for communications, networks and systems & workshops*. ICST, p.72.
21. Ling TC, Lee JF and Hoh KP. Reducing handoff delay in wlan using selective proactive context caching. *Malays J Comput Sci* 2010; 23(1): 49–59.
22. Le Y, Ma L, Cheng W, et al. A time fairness-based mac algorithm for throughput maximization in 802.11 networks. *IEEE Trans Comput* 2013; 64(1): 19–31.
23. Khayam SA and Radha H. 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, pp.100–107.
24. Kamthe A, Carreira-Perpinán MA and Cerpa AE. M&M: Multi-level Markov model for wireless link simulations. In: *Proceedings of the 7th ACM conference on embedded networked sensor systems*. ACM, pp.57–70.

25. Ghosh C, Cordeiro C, Agrawal DP, et al. Markov chain existence and hidden Markov models in spectrum sensing. In: *IEEE international conference on pervasive computing and communications, 2009. PerCom 2009*. IEEE, pp.1–6.
26. Prasad PS and Agrawal P. Movement prediction in wireless networks using mobility traces. In: *2010 7th IEEE consumer communications and networking conference (CCNC)*. IEEE, pp.1–5.
27. Cheikh AB, Ayari M, Langar R, et al. Optimized handoff with mobility prediction scheme using hmm for femtocell networks. In: *2015 IEEE international conference on communications (ICC)*. IEEE, pp.3448–3453.
28. Kashyap A, Paul U and Das SR. Deconstructing interference relations in wifi networks. In: *2010 7th annual IEEE communications society conference on sensor mesh and ad hoc communications and networks (SECON),*. IEEE, pp.1–9.
29. Paul U, Kashyap A, Maheshwari R, et al. Passive measurement of interference in wifi networks with application in misbehavior detection. *IEEE Trans Mobile Comput* 2013; 12(3): 434–446.
30. Reynolds D. *Universal Background Models*. Boston: Springer US, 2015. pp.1547–1550.
31. Gupta M, Gao J, Aggarwal CC, et al. Outlier detection for temporal data: A survey. *IEEE Trans Knowl Data Eng* 2014; 26(9): 2250–2267.
32. Zhang D, Gatica-Perez D, Bengio S, et al. Semi-supervised adapted hmms for unusual event detection. In: *IEEE computer society conference on computer vision and pattern recognition, 2005. CVPR 2005*, volume 1. IEEE, pp.611–618.
33. OMNeT++ Discrete Event Simulator. https://www.omnetpp.org/. Accessed June 2020.
34. INET Framework. https://inet.omnetpp.org/. Accessed June 2020.
35. Fu B, Bernáth G, Steichen B, et al. Wireless background noise in the Wi-Fi spectrum. In: *4th international conference on wireless communications, networking and mobile computing, 2008. WiCOM'08*. IEEE, pp.1–7.
36. INET Framework Documentation. https://inet.omnetpp.org/docs/index.html. Accessed June 2020.
37. R: What is R. https://www.r-project.org/about.html. Accessed June 2020.

Author biographies

Anisa Allahdadi received a BSc and MSc in computer science and software engineering, respectively, from BIHE University (Bahá'í Institute for Higher Education) in Iran. Anisa holds a PhD degree in Computer Science from the University of Porto under the MAP-i Doctoral Programme. She is currently a researcher in the Center of High-Assurance Software at INESC TEC. Her research interests include machine learning, data mining, deep learning, natural language processing, probabilistic graphical models, and wireless network management and security.

Ricardo Morla is an assistant professor at the University of Porto. His research interests are in network security and AI, mostly looking at sidechannel attacks on encrypted traffic for privacy protection and for malware C2 traffic detection. He tries to understand the adversarial nature and the AI on big data challenges of these attacks. Ricardo teaches and does research at the Electrical and Computer Engineering Department at FEUP and at INESC TEC. He holds a PhD in Computing from Lancaster University. He was a lecturer and post-doc at UC Irvine in 2007, and a visiting faculty at Carnegie Mellon University in 2010 under the CMU-Portugal program. He currently runs FEUP's Network Lab.

Jaime S Cardoso, Senior Member, IEEE, received the Licenciatura (5 year) degree in electrical and computer engineering, the MSc degree in mathematical engineering, and the PhD degree in computer vision from the University of Porto in 1999, 2005, and 2006, respectively. He is currently an Associate Professor with Habilitation with the Faculty of Engineering, University of Porto and the Coordinator of the Centre for Telecommunications and Multimedia, INESC TEC. From 2012 to 2015, he served as the President of the Portuguese Association for Pattern Recognition (APRP), affiliated in the International Association for Pattern Recognition (IAPR). His research can be summed up in three major topics computer vision, machine learning, and decision support systems. Cardoso has co-authored 250 + papers, 80 + of which in international journals. The research results have been recognized both by the peers, with 4500 + citations to his publications and the advertisement in the mainstream media several times.