Big Data Analytics for Intrusion Detection System: Statistical Decision-Making Using Finite Dirichlet Mixture Models

Nour Moustafa, Gideon Creech, and Jill Slay

Abstract An intrusion detection system has become a vital mechanism to detect a wide variety of malicious activities in the cyber domain. However, this system still faces an important limitation when it comes to detecting zero-day attacks, concerning the reduction of relatively high false alarm rates. It is thus necessary to no longer consider the tasks of monitoring and analysing network data in isolation, but instead optimise their integration with decision-making methods for identifying anomalous events. This chapter presents a scalable framework for building an effective and lightweight anomaly detection system. This framework includes three modules of capturing and logging, pre-processing and a new statistical decision engine, called the Dirichlet mixture model based anomaly detection technique. The first module sniffs and collects network data while the second module analyses and filters these data to improve the performance of the decision engine. Finally, the decision engine is designed based on the Dirichlet mixture model with a lower-upper interquartile range as decision engine. The performance of this framework is evaluated on two well-known datasets, the NSL-KDD and UNSW-NB15. The empirical results showed that the statistical analysis of network data helps in choosing the best model which correctly fits the network data. Additionally, the Dirichlet mixture model based anomaly detection technique provides a higher detection rate and lower false alarm rate than other three compelling techniques. These techniques were built based on correlation and distance measures that cannot detect modern attacks which mimic normal activities, whereas the proposed technique was established using the Dirichlet mixture model and precise boundaries of interquartile range for finding small differences between legitimate and attack vectors, efficiently identifying these attacks.

N. Moustafa (✉) • G. Creech • J. Slay
The Australian Centre for Cyber Security, University of New South Wales Canberra,
Canberra, NSW, Australia
e-mail: nour.moustafa@unsw.edu.au; G.Creech@adfa.edu.au; j.slay@adfa.edu.au

I. Palomares et al. (eds.), *Data Analytics and Decision Support for Cybersecurity*,
Data Analytics, DOI 10.1007/978-3-319-59439-2_5

1 Introduction

In the cyber security field, an Intrusion Detection System (IDS) is essential for achieving a solid line of defence against cyber intrusions. The digital world has become the principal complement to the physical world because of the widespread usage of computer networks and prevalence of programs and services which easily accomplish users' tasks in a short time at a low cost. A system is considered secure if the three principles of computer security, Confidentiality, Integrity and Availability (CIA), are successfully satisfied [38, 43]. Hackers always endeavour to violate these principles, with each attack type having its own sophisticated manner and posing serious threats to computer networking.

An Anomaly-based Detection System (ADS), a specific methodology of IDS discussed in Sect. 2, still faces two problems for implementation in large-scale industrial applications [2, 7, 39], such as cloud computing [30] and SCADA systems [16]. Firstly, and most importantly, the construction of a profile from various legitimate patterns is extremely difficult because of the frequent changes of the normal data [2, 34, 47]. Secondly, the process of building a scalable, adaptive and lightweight detection method is an arduous task with the high speeds and large sizes of current networks [39, 47].

ADS methodologies have been developed using approaches involving data mining and machine learning, artificial intelligence, knowledge-based and statistical models [7, 34, 39]. Nevertheless, usually, these proposed techniques have produced high False Positive Rates (FPRs) because of the difficulty of designing a solution which solves the above problems. Recent research studies [27, 34, 41, 44, 47] have focused on statistical models due to the ease of concurrently employing and determining the potential properties of both normal and abnormal patterns of behaviour. Discovering these properties and characterising a certain threshold for any detection method to correctly detect attacks requires accurate analysis.

Both network and host systems have multiple devices, software, sensors, platforms and other sources connected together to deliver services to users and organisations anytime and anywhere. In addition, these systems monitor the demands from such organisations by using Big Data analytical techniques and tools to carefully provide decision support for distinguishing between normal and anomalous instances. For these reasons, the capture and processing of these data are dramatically increasing in terms of 'volume' , 'velocity' and 'variety' , which are referred to as the phenomena of 'Big Data' [53]. The Big Data paradigm poses a continuous challenge in the use of network or host data sources for the design of an effective and scalable ADS.

Monitoring and analysing network traffic have attained growing importance for several reasons. Firstly, they increase visibility to the user, system and application traffic by gathering and analysing network flow records which also helps to track the bandwidth consumption of users and systems to ensure robust service delivery. Secondly, they identify performance bottlenecks and minimising non-business bandwidth consumption. Thirdly, there are advantages related to IDS technology, which are the tracking of network traffic using protocol analysis for recognising potential attack profiles, such as UDP spikes. Finally, they monitor network traffic, peer-to-peer protocols and URLs for a specific device or network to determine suspicious activities and unauthorised access [5].

In the literature, if data does not fit a normal distribution, it will be better to fit and detect outliers/anomalies using mixture models, especially Gaussian Mixture Model (GMM), Beta Mixture Model (BMM) or Dirichlet Mixture Model (DMM) [15, 17, 34, 50]. According to [17, 18, 21], the DMM can fit and define the boundaries of data better than other mixture models because it consists of a set of probability distributions. Moreover, the DMM is more suitable for modelling streaming data, for example, data originating from videos, images, or network traffic. The mathematical characteristics of the DMM also permit the representation of samples in a transformed space in which features are independent and identically distributed (i.i.d.). In the case of high dimensionality, the DMM for clustering data provides higher accuracy than other mixture models [9]. Therefore, we use this model to properly fit network data using the lower-upper Interquartile Range (IQR) [40] as a threshold to detect any observation outside them as an anomaly.

In this chapter, we propose a scalable framework for building an effective and lightweight ADS that can efficiently identify suspicious patterns over network systems. The framework consists of a capturing and logging module to sniff and record data, a pre-processing module to analyse and filter these data and the proposed ADS statistical decision engine, based on the DMM, for recognising abnormal behaviours in network systems.The DMM model is a statistical technique developed based on the method of anomaly detection which computes the density of Dirichlet distributions for the normal profile (i.e., the training phase) and testing phase (using the parameters estimated from the training phase). The decision-making method for identifying known and new anomalies is designed by specifying a threshold of the lower-upper IQR for the normal profile and considering any deviation from it as an attack.

The performance of this framework is evaluated on two well-known datasets, the NSL-KDD[1], which is an improved version of the KDD99 and the most popular dataset used for evaluating IDSs [48], and our UNSW-NB15[2] which involves a wide variety of contemporary normal, security and malware events [34]. The Dirichlet mixture model based anomaly detection technique is compared with three recent techniques, namely the Triangle Area Nearest Neighbours (TANN) [49], Euclidean Distance Map (EDM) [46] and Multivariate Correlation Analysis (MCA) [47]. These techniques were developed based on computing distances and correlations between legitimate and malicious vectors, which cannot often find a clear difference between these vectors, especially with modern attack styles that mimic normal ones.

[1] The NSLKDD dataset, https://web.archive.org/web/20150205070216/http://nsl.cs.unb.ca/NSL-KDD/, November 2016.

[2] The UNSW-NB15 dataset, https://www.unsw.adfa.edu.au/australian-centre-for-cyber-security/cybersecurity/ADFA-NB15-Datasets/, November 2016.

However, our technique was established using the methods of the Dirichlet mixture model and the accurate boundaries of interquartile range that properly find the small differences between these vectors, considerably improving the detection accuracy.

The key contributions of this chapter are as follows.

1. We propose a new scalable framework for anomaly detection system on large-scale networks. In this framework, we also develop a novel decision engine based on the Dirichlet Mixture Model and lower-upper Interquartile Range to efficiently detect malicious events.
2. We describe how statistical analysis can define the normality of network data to choose the proper model that correctly fits the data and make an intelligent decision-making method which discriminates between normal and abnormal observations.
3. A performance evaluation of this framework is conducted on two benchmark datasets: the NSL-KDD which is the most common dataset and UNSW-NB15 which is the latest dataset used for assessing IDSs, as well as comparing this technique with three existing techniques to assess its reliability for detecting intrusions.

The rest of this chapter is organised as follows. The background on Intrusion and Anomaly Detection Systems are presented in Sect. 2. Section 3 describes related work on decision engine approaches. The DMM-based technique is explained in Sect. 4 and the proposed scalable framework discussed in Sect. 5. The experimental results and discussions are provided in Sect. 6. Finally, concluding remarks are presented in Sect. 7.

2 Background on Intrusion and Anomaly Detection Systems

An Intrusion Detection System (IDS) is defined as a technique for monitoring host or network activities to detect possible threats by measuring their violations of Confidentiality, Integrity and Availability (CIA) principles [6, 42, 43]. There are two types of IDSs depending on the data source: a host-based IDS monitors the events of a host by collecting information about activities which occur in a computer system [51] while a network-based IDS monitors network traffic to identify remote attacks that take place across that network [11, 38].

IDS methodologies are classified into three major categories: Misuse-based (MDS); Stateful Protocol Analysis (SPA); and ADS [29, 38]. A MDS monitors host or network data audits to compare observed behaviours with those on an identified blacklist. Although it offers relatively high Detection Rates (DRs) and low FPRs, it cannot identify any zero-day attack. Also, it requires a huge effort to regularly update the blacklist which is a set of rules for each malicious category generated by security expertise [47]. A SPA inspects protocol states, especially a pair of request-response protocols, for example, HTTP ones. Although it is quite similar to an ADS, it depends on vendor-developed profiles for each protocol. Finally, an ADS establishes a normal profile from host or network data and discovers any variation from it as an attack. It can detect existing and new attacks, as it does not require any effort to generate rules, an ADS has become a better solution than MDS and SPA [13, 34, 38, 43]. However, it still has some challenges, as explained in Sect. 2.2, that we will try to mitigate.

An IDS's deployment architecture is classified as either distributed or centralised. A distributed IDS is a compound system involving several intrusion detection subsystems installed at different locations and connected in order to transfer relevant information. In contrast, a centralised IDS is a non-compound system deployed at only one location, with its architecture dependent on an organisation's size and sensitivity of its data which should be considered in terms of its deployment [28].

2.1 ADS Components

A typical ADS contains four components, a data source, a data pre-processing module, a decision engine technique and a defense response module [13, 34], as depicted in Fig. 1 and described in detail below.

- **Data source module**—This component is an essential part of any ADS that provides the potential host or network audit data to enable the DE to classify observations as either normal or attack [33]. Several data sources have been collected in offline datasets, such as the KDD CUP 99, NSL-KDD and UNSW-NB15, which consist of a wide variety of normal and malicious records for evaluating the performance of DE approaches. With the high speeds and large sizes of current communication systems, each data source has big data terms, i.e., a large volume, velocity and variety. Therefore, it is vital to design an effective and scalable ADS which can handle such data and make the correct decision upon the detection of malicious observations.

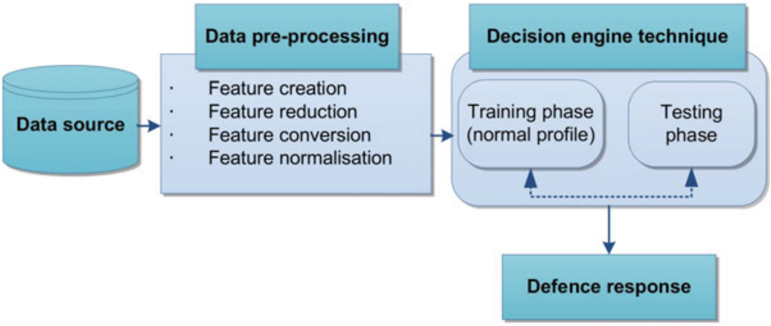

Fig. 1 Components of ADS

- **Data pre-processing module**—Data pre-processing is an important stage in any framework involving learning from data, as it handles and filters the input audit data by removing duplicated, irrelevant and/or noisy features to create a set of patterns that are then passed to the DE with the aim of improving the performance of that DE for detecting anomalous activity. It includes the following set of functions.
  - **Feature creation**—Feature creation constructs a set of features/attributes from host or network data using different tools, for instance, BRO-IDS, Netflow and Netmate. It is impossible to operate an IDS on raw data without mining a subset of features, such as the attributes in the NSL-KDD and UNSW-NB15 datasets.
  - **Feature reduction**—The function of feature reduction is to exclude unnecessary and duplicated attributes, and can be divided into feature selection and feature extraction methods. The first finds a subset of the original features and the second transforms the data from a high- to lower-dimensional space, such as using Principal Component Analysis (PCA) [12].
  - **Feature conversion**—The function of feature conversion is to convert feature types, which can be numeric or symbolic, into numeric values for ease of use in decision engine approaches as data analytics and statistical decision engine cannot use symbolic features to define data patterns.
  - **Feature normalisation**—Feature normalisation is a measure for scaling data features into a particular range, for example, [0,1], and is important for eliminating bias from raw data without modifying the statistical properties of the attributes.
- **DE module**—Intuitively, the decision engine is responsible for a critical stage, which is the design of an effective and efficient system for discovering intrusive activities in large-scale data in real time. Selecting the appropriate functions for a DE approach, and its training and testing phases, contributes to measuring the effectiveness of an IDS as, if it is not performed properly, the overall protection level will be easily compromised.
- **Security response module**—This module aims at indicating and explaining a decision taken by the system or administrators to prevent attack activities. More specifically, if a malicious event is detected, an alert will be raised to the security administrator for preventing this event.

2.2 ADS Challenges

Although a MDS cannot recognise zero-day attacks or even variants of existing attacks, it is still a common defence solution used in commercial products. On the contrary, an ADS can detect serious threats but has often been faced with potential challenges for its effective design. These challenges could be explored using an anomaly-based method, which is the construction of a purely normal profile with any variation from it declared an anomaly [24, 30, 37, 53], as follows.

- Establishing a profile which includes all possible normal patterns is very complex to achieve, as the boundary between normal and suspicious activities is always inaccurate. There are False Positive Rate (FPR) and False Negative Rate (FNR) errors which occur when a normal behaviour falls in an attack region (i.e. it is classified as an attack) and a malicious behaviour in a normal region, respectively.
- When designing the architecture of an adaptive and scalable ADS, it requires a careful analysis to discriminate attacks from the normal profile as sophisticated malicious activities, such as stealth and spy attacks [20], can adapt to be almost the same as normal patterns. Consequently, methods for detecting them have to analyse and inspect the potential characteristics of the network traffic.
- Real-time detection is also very challenging for reasons which increase its processing time and false alarm rate if not properly addressed. Firstly, the features created for network traffic may contain a set of noisy or irrelevant features. Secondly, the lightweight detection methods need to be carefully adopted, with respect to the above problems.
- Obtaining a decent-quality dataset is usually a major concern for evaluating, learning and validating ADS models. It should have a wide range of modern normal and malicious observations as well as being correctly labelled. This requires a huge effort of analysing the data in order to ensure establishing an authentic truth table, which has the security events and malware for the correct labelling process.
- The deployment of an ADS architecture is often difficult in large-scale environments, in particular, cloud computing and SCADA systems, have multiple nodes which could be either centralised or distributed. Also, the high speeds and a large amount of data transferring between these nodes often affect the performance of an ADS.

2.3 ADS Deployment in Enterprise Systems

With the new era of the Internet of Things (IoT), which is the networked interconnections of everyday objects often associated with their ubiquitous use, many applications and systems need to be protected against intrusive activities. As cloud computing environments and Supervisory Control and Data Acquisition (SCADA) systems are currently fully dependent on the Internet, they require an adaptable and scalable ADS for identifying the malicious events they frequently face. Cloud computing is a “network of networks” based on Internet services in which virtual shared servers provide the software, platform, infrastructure and other resources [3]. It consists of the three service models Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) [30]. To detect attacks, an

IDS can be installed on a virtual server as a host-based IDS or deployed across the network as a network-based IDS or, by configuring both, provide a better defence.

SCADA systems monitor and control industrial and critical infrastructure functions, for example, water, electricity, railway, gas and traffic [16]. Like the cloud computing environment, with the rapid increase in the Internet and interconnected networks, these systems face complicated attacks, such as DoS and DDoS, which highlight the need for stronger SCADA security. However, designing the architecture and deployment of an adaptive and scalable ADS for these environments has become a big challenge because the high speeds and large sizes of existing networks generate a massive number of packets each time, and those packets should be inspected simultaneously in order to identify malicious activities.

3 Related Work on Decision Engine Approaches

Many researchers have investigated decision engine approaches, which can be categorised into five types: classification-based approaches [7, 10, 13]; clustering-based approaches [2, 7, 13, 17]; knowledge-based approaches [7, 10, 13]; combination-based approaches [7, 11, 13, 52]; and statistical-based approaches [7, 13, 34, 47], as illustrated in Table 1. Firstly, classification is a way of categorising data observations in particular classes in a training set with a testing set containing other instances for validating these classes; for instance, Horng et al. [23] proposed a Network-based ADS which included a hierarchical clustering and support vector machine to reduce the training time and improve detection accuracy. Ambusaidi et al. [4] developed a least-square support vector machine for the design of a lightweight Network-based ADS by selecting the significant features of network data and detecting anomalies. Recently, Dubey et al. [14] developed a Network-based ADS based on the collection techniques of an artificial neural network, k-means and Naïve-Bayes to improve the detection of malicious activities. However, overall, classification-based IDSs rely heavily on the assumption that each classifier has to be adjusted separately and always consume more resources than statistical techniques.

Secondly, clustering involves unsupervised machine-learning algorithms which allocate a set of data points to groups based on the similar characteristics of the points, for example, distance or probability measures. Nadiammai et al. [35] analysed and evaluated k-means, hierarchical and fuzzy c-means clustering techniques for building a Network-based ADS and reported that the complexity and detection accuracy of the fuzzy c-means algorithm were better than those of the others. Jadhav et al. [25] proposed a Network-based ADS based on clustering network packets and developed a new data pre-processing function using the fuzzy logic technique for classifying the severity of attacks in network traffic data. Zainaddin et al. [52] proposed a hybrid of fuzzy clustering and an artificial neural network to construct a Network-based ADS which efficiently detected malicious events. Clustering-based ADS techniques have several advantages. Firstly, they group data points in an unsupervised manner which means that they do not need to provide

Table 1 Comparison of decision engine approaches

Decision engine approaches	Related works	Advantages	Disadvantages
Classification	Horng et al. [23], Ambusaidi et al. [4], Dubey et al. [14]	• Provide higher detection rate and lower false positive rate if the network data is correctly labelled	• Depend on the assumption that each classifier has to be built separately • Consume more computational resources
Clustering	Nadiammai et al. [35], Jadhav et al. [25], Zainaddin et al. [52]	• Group data with no need to the class label • Decrease processing times	• Rely on the efficiency of building a normal profile • Require a higher time while updating this profile
Knowledge	Naldurg et al. [36], Hung et al. [24]	• Discriminate existing attacks • Provide higher detection rate	• Take too much time during the processing • Use static rules for defining malicious patterns
Combination	Perdisci et al. [37], Aburomman et al. [1], Shifflet [45]	• Achieve higher accuracy and detection • Demand only a set of controlling parameters to be adjusted	• Need a huge effort to integrate some techniques • Take a long processing time than other techniques
Statistics	Fan et al. [17], Greggio [18], Zhiyuan et al. [47]	• Accomplish higher accuracy and detection if a baseline of identifying attacks correctly adapted • Do not consume resources like other techniques	• Require accurate analysis to select the correct baseline • Need new functions to define attack types

class labels for observations. Secondly, they are effective for grouping large datasets into similar groups to detect network anomalies. However, in contrast, clustering is highly dependent on the efficacy of constructing a normal profile and the difficulty of automatically updating it.

Thirdly, knowledge-based methods establish a set of patterns from input data to classify data points with respect to class labels, with common knowledge-based ADSs rule-based, expert systems and ontologies. Naldurg et al. [36] suggested a framework for intrusion detection using temporal logic specifications with intrusion patterns formulated in a logic structure called EAGLE. It supported data values and parameters in recursive equations and enabled the identification of intrusions with temporal patterns. Hung et al. [24] presented an ontology-based approach for establishing a Network-based ADS according to the end-users' domain in which, as ontologies were applied as a conceptual modelling technique, a Network-based ADS could be simply built. Knowledge-based algorithms have some advantages. They are sufficiently robust and flexible to detect existing attacks in a small-scale system and achieve a high DR if a significant knowledge base about normal and abnormal instances can be correctly extracted. Conversely, they have FPRs due to the unavailability of biased normal and intrusion audit data and cannot identify rare or zero-day anomalies, and dynamically updating their rules is difficult.

Fourthly, combination-based techniques use many methods to effectively classify data instances, with most used for ADSs ensemble- and fusion-based techniques; for instance, Perdisci et al. [37] established a high-speed payload Network-based ADS based on an ensemble of one-class support vector machine for improving detection accuracy. Aburomman et al. [1] suggested an ensemble method which used PSO-generated weights to build a hybrid of more accurate classifiers for a Network-based ADS created based on local unimodal sampling and weighted majority algorithm approaches to improve the accuracy of detecting attacks. Shifflet [45] discussed a platform which enabled a hybrid of classification techniques to be executed together to build a fusion mechanism for the state of a network that was capable of efficiently detecting anomalous activities. Combination-based methods are advantageous as they achieve higher accuracy and detection rate than single ones while requiring a set of controlling parameters that can be easily adjusted. However, adopting a sub-set of consistent and unbiased classification techniques is difficult because it depends on using a hybridisation measure to combine them. Also, it is evident that their computational costs for large amounts of network traffic data are high due to the number of classifiers used.

Finally, in statistical-based approaches, an anomaly is a rare event which occurs among natural data ones and is measured by statistical methods which could be of the first order, such as means and standard deviations, second order, such as correlation measures, or third order, such as hypothesis testing and mixture models; for example, Fan et al. [17] developed an unsupervised statistical technique for identifying network intrusions in which legitimate and anomalous patterns were learned through finite generalised Dirichlet mixture models based on a Bayesian inference, with the parameters of the models and saliency of features simultaneously estimated. Greggio [18] designed a Network-based ADS based on the unsupervised fitting of network data using a Gaussian Mixture Model which selected the number of mixture components and fit the parameter for each component in a real environment. They extended their study to provide an efficient method for the varied learning of finite Dirichlet mixture models to design a Network-based ADS. This approach was based on the establishment and optimisation of a lower boundary for the likelihood of the model by adopting factored conditional distributions through its variables.

Overall, although ADS statistical-based approaches can analyse and determine the potential characteristics of normal and abnormal observations, identifying them and defining a certain baseline which distinguishes between normal and abnormal instances need accurate analysis. Therefore, we propose the methodology of the Dirichlet mixture model with the precise boundaries of interquartile range function as a decision engine. This is one of the statistical approaches that can define the inherent patterns of both legitimate and malicious features and observations, finding a clear variation between these observations. However, the other approaches often depend on many internal processes with a kernel function(s) that have to be adjusted for each problem. The main motivation for selecting this methodology is that statistical analytics of network data have shown that these data do not belong to a Gaussian distribution [17, 34]. Therefore, it is better to apply non-Gaussian distributions, such as Dirichlet mixture model to correctly fit network data using the lower-upper interquartile range function to detect any observation outside this range as an outlier/anomaly.

4 DMM-Based ADS Technique

This section describes the mathematical aspects of estimating and modelling data using the DMM, and discusses the proposed methodology for using this model to build an effective ADS.

4.1 Finite Dirichlet Mixture Model

Because a finite mixture model can be considered a convex combination of two or more Probability Density Functions (PDFs), the joint properties of which can approximate any arbitrary distribution, it is a powerful and flexible probabilistic modelling tool for handling multivariate data, such as network data [54]. A finite mixture of Dirichlet distributions with K components is shown in Fig. 2 and is given by [8, 18]

$$p(X|\pi, \alpha) = \sum_{i=1}^{K} \pi_i Dir(X|\alpha_i) \tag{1}$$

where $\pi = (\pi_1, \ldots, \pi_K)$ refers to the mixing coefficients, which are positive, with their summation 1, $\sum_{i=1}^{K} \pi_i$,$\alpha = (\alpha_1, \ldots, \alpha_K)$, and $Dir(X|\alpha_i)$ indicates the Dirichlet distribution of component i with its own positive parameters ($\alpha = (\alpha_{i1}, \ldots, \alpha_{iS})$) as

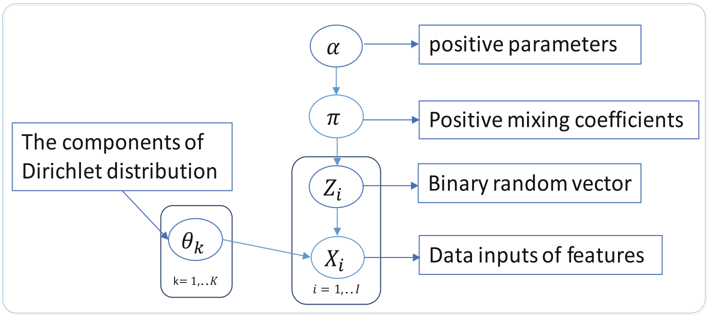

Fig. 2 Finite mixture model

$$Dir(X|\alpha_i) = \frac{\Gamma(\sum_{s=1}^{S} \alpha_{is})}{\prod_{s=1}^{S} \Gamma(\alpha_{is})} \prod_{s=1}^{S} X_s^{\alpha_{is}-1} \tag{2}$$

where $X = (X_1, \ldots, X_S)$, S is the dimensionality of X and $\sum_{s=1}^{S} x_s = 1, 0 \leq X_s \geq 1$ for $s = 1, \ldots, S$. It is worth noting that a Dirichlet distribution is used as a parent distribution to directly model the data rather than as a prior to the multinomial.

Considering a set of N independent identically distributed ($i.i.d$) vectors ($X = \{X_1, \ldots, X_N\}$) assumed to be produced from the mixture distribution in Eq. (1), the likelihood function of the DMM is

$$p(X|\pi, \alpha) = \prod_{l=1}^{N} \{\sum_{i=1}^{K} \Pi_i Dir(X_l|\alpha_i)\} \tag{3}$$

The finite mixture model in Eq. (1) is considered as a latent variable model. Therefore, for each vector (X_i), we introduce a K-dimensional binary random vector ($Zi = \{Z, \ldots, Z_{iK}\}$), where $Z_{is} \in \{0, 1\}$, $\sum_{i=1}^{K}$ and $Z_{is} = 1$ if X_i belongs to component i, otherwise 0. For the latent variables ($Z = \{Z_1, \ldots, Z_N\}$), which are actually hidden ones that do not appear explicitly in the model, the conditional distribution of Z given the mixing coefficients (π) is defined as

$$p(Z|\pi) = \prod_{l=1}^{N} \prod_{i=1}^{K} \pi_i^{Z_{li}} \tag{4}$$

Then, the likelihood function with the latent variables, which is actually the conditional distribution of a dataset L given the class labels, Z can be written as

$$p(X|\pi, \alpha) = \prod_{l=1}^{N} \prod_{i=1}^{K} Dir(X_l|\alpha_i) \tag{5}$$

Given a dataset L has a set of features D, an important problem is the learning process of the mixture parameters, that is, both estimating the parameters and selecting the number of components (K). In order to estimate these parameters and select the number of components correctly, we apply the Maximum Likelihood (ML) proposed in [30]. We suggest a new DMM for designing an ADS, namely DMM-ADS, in which includes training and testing phases for learning and validating network data. In the training phase, the DMM parameters and IQR are estimated to construct a normal profile, with abnormal instances identified in the testing phase, as detailed in the following two sections.

4.2 Training Phase of Normal Instances

The construction of a purely normal training set is extremely vital to ensure correct detection. Given a set of normal instances $(r_{1:n}^{normal})$ in which each record comprises a set of features D, where $r_{1:n}^{normal} = \{x_1, x_2, \ldots, x_D\}^{normal}$, the normal profile includes only statistical measures from $r_{1:n}^{normal}$. They include the estimated parameters (π, α, Z) of the DMM to compute the PDF of the Dirichlet distribution $(Dir(X|\pi, \alpha, Z))$ for each observation in the training set.

Algorithm 1 involves the proposed steps for establishing a normal profile (pro), with the parameters (π, α, Z) of the DMM computed for all the normal observations $(r_{1:n}^{normal})$ using the equations published in [30], and then the PDFs of the attributes $(X_{1:D})$ calculated using Eqs. (1)–(5). Next, the IQR is computed by subtracting the first quartile from the third quartile of the PDFs [40] to establish a threshold for recognising abnormal instances in the testing phase. It is acknowledged that quantiles are dividing a range of data into contiguous intervals with equal probabilities [40].

Algorithm 1: Normal profile construction of normal instances

```
Input: normal instances $(r_{1:n}^{normal})$
Output: normal profile (pro)
for each record i in $(r_{1:n}^{normal})$ do
    estimate the parameters $(\pi_i, \alpha_i, Z_i)$ of the DMM as in [29]
    calculate the PDFs using equations (1) to (5) based on the estimated parameters of
    Step 2
end for
compute $lower = quantile(PDFs, 1)$
compute $upper = quantile(PDFs, 3)$
compute $IQR = upper - lower$
pro $\leftarrow ((\pi, \alpha_i, Z_i), (lower, upper, IQR))$
return pro
```

Algorithm 2: Testing phase and decision-making method

```
Input: observed instance ($r^{testing}$), pro
Output: normal or attack
calculate the PDFtesting using equations using the parameters $(\pi_i, \alpha_i, Z_i)$
if $(PDF^{testing} < (lower–w * (IQR)) || (PDF^{testing} > (upper + w * (IQR))$ then
    return attack
else
    return normal
end if
```

4.3 Testing Phase and Decision-Making Method

In the testing phase, the Dirichlet PDF ($PDF^{testing}$) of each observed record ($r^{testing}$) is computed using the same parameters estimated for the normal profile $((\pi, \alpha_i, Z_i)$,$(lower, upper, IQR))$. Algorithm 2 includes the steps in the testing phase and decision-making method for identifying the Dirichlet PDFs of the attack records, with step 1 constructing the PDF of each observed record using the stored normal parameters (π_i, α_i, Z_i).

Steps 2 to 6 define the decision-making process. The IQR of the normal instances is computed to find the outliers/anomalies of any observed instance ($r^{testing}$) in the testing phase which are considered to be any observations falling below $(lower - w * (IQR))$ or above $(upper + w * (IQR))$, where w indicates the interval values between 1.5 and 3 [40]. The detection decision is based on considering any $PDF^{testing}$ falling out of this range as an attack record, otherwise normal.

5 Scalable ADS Framework

This section discusses a proposed scalable framework for developing an effective ADS which identifies malicious activities in large-scale environments. It consists of three modules, capturing and logging, data pre-processing and a DMM-based ADS technique, as shown in Fig. 3.

In the first phase, a set of attributes is created from network traffic to capture network connections for a particular time window. It is observed that the best way of analysing network data is to sniff the traffic from the router located at each network node and aggregate only relevant network packets [46, 47]. Secondly, the pre-processing step filters network data by converting symbolic features into numeric ones, as shown in Fig. 4.

The reason for this conversion process is because statistical methods, such as the DMM-based ADS technique, can handle only numeric data. Furthermore, selecting the most significant features to improve the performance and reduce the processing time of the decision engine in order to deploy it in real time. Finally, we propose the

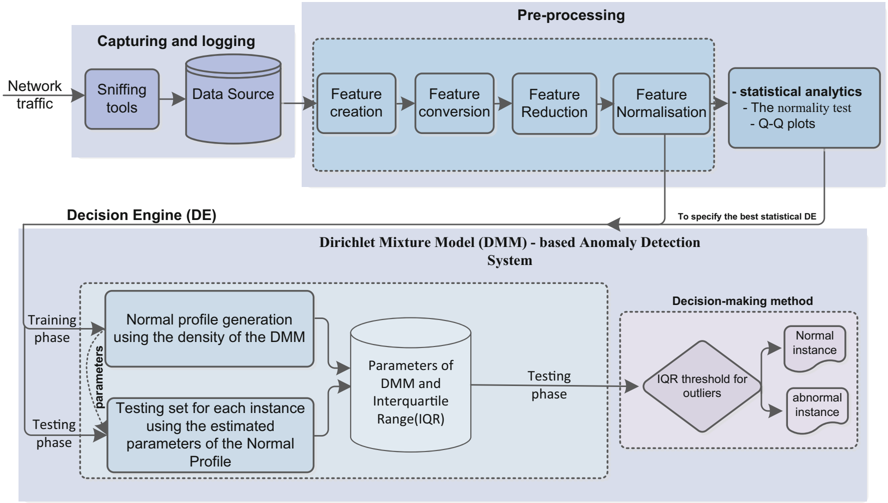

Fig. 3 Proposed scalable framework for designing effective ADS

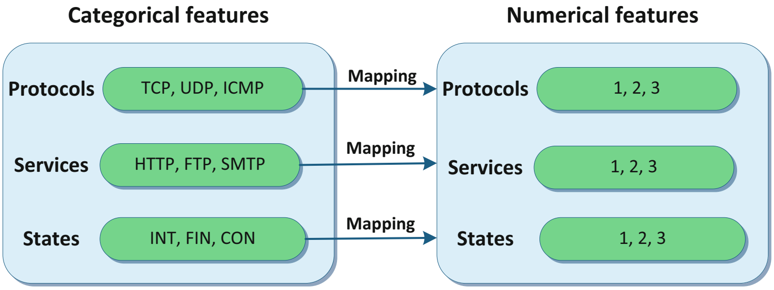

Fig. 4 Example of converting categorical features into numerical features using UNSW-NB15 dataset

DMM-based ADS technique as the decision engine, which efficiently discriminates between legitimate and suspicious instances, as discussed in the previous section.

5.1 *Capturing and Logging Module*

This module sniffs network data and stores them to be processed for the decision engine, like the steps for creating our UNSW-NB15 dataset [33, 34]. An IXIA PerfectStorm tool,[3] which has the capability to determine a wide range of network

[3] The IXIA Tools, https://www.ixiacom.com/products/perfectstorm, November 2016.

Fig. 5 Functions of IXIA PerfectStorm tool

segments and elicits traffic for several web applications, such as Facebook, Skype, YouTube and Google, was used to mimic recent realistic normal and abnormal network traffic, as shown in Fig. 5. It could also simulate the majority of security events and malicious scripts which is difficult to achieve using other tools. The configuration of the UNSW-NB15 testbed was used to simulate a large-scale network and a Tcpdump tool to sniff packets from the network's interface while Bro, Argus tools and other scripts were used to extract a set of features from network flows.

In [33, 34], the UNSW-NB15 was created, which comprises a wide variety of features. These features can be classified into packet-based and flow-based. The packet based features help in examining the packet payload and headers while the flow based features mine information from the packet headers, such as a packet direction, an inter-arrival time of packets, the number of source/destination IPs for a particular time window, and an inter-packet length. AS depicted in Fig. 6, the pcap files[4] of this dataset were processed by the BRO-IDS and Argus tools to mine the basic features. Then, we developed a new aggregator module to correlate its flows. These flows were aggregated for each 100 connections, where packets with the same source/destination IP addresses and ports, timestamp, and protocol were collected in a flow record [31, 32]. This module enables to establish monitoring applications for analysing network characteristics such as capacity, bandwidth, rare and normal events.

[4] **Pcap** refers to **p**acket **cap**ture, which contains an Application Programming Interface (API) for saving network data. The UNIX operating systems execute the pcap format using the libpcap library while the Windows operating systems utilise a port of libpcap, called WinPcap.

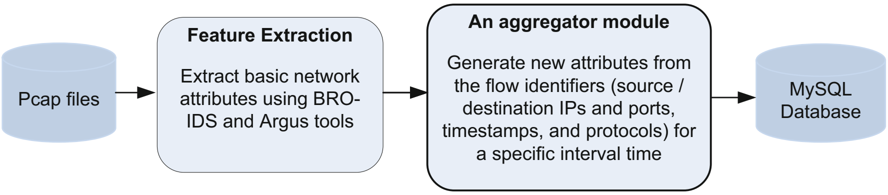
Pcap files
Feature Extraction
Extract basic network attributes using BRO-IDS and Argus tools
An aggregator module
Generate new attributes from the flow identifiers (source / destination IPs and ports, timestamps, and protocols) for a specific interval time
MySQL Database

Fig. 6 Extracting and aggregating features of UNSW-NB15 dataset

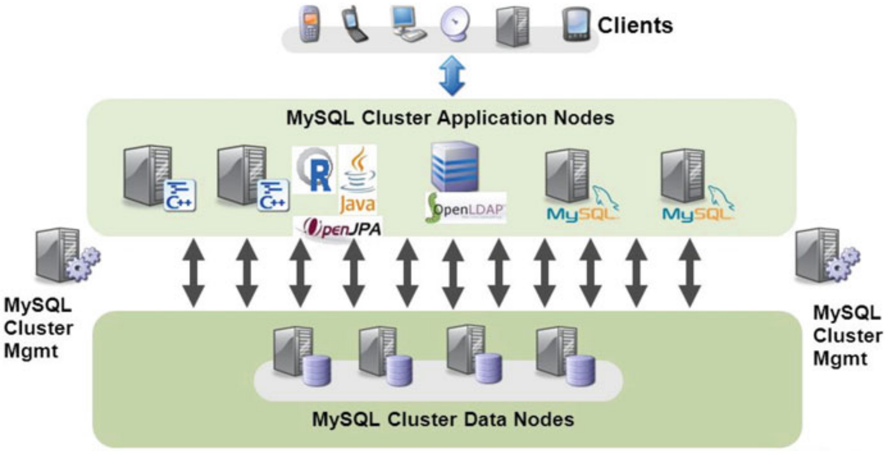

Fig. 7 Architecture of MySQL cluster CGE

These features were recorded using the MySQL Cluster CGE technology[5] that has a highly scalable and real-time database, enables a distributed architecture to read and write intensive workloads and is accessed via SQL or NoSQL APIs, as depicted in Fig. 7. It can also support memory-optimised and disk-based tables, automatic data partitioning with load balancing, and can add nodes to a running cluster for handling online big data. Although this technology has a similar architecture to Hadoop tools[6], which are the most popular for processing big offline data, an ADS has to detect malicious behaviours in real time. These features are then passed to the pre-processing module to be analysed and filtered.

[5] The MySQL Cluster CGE technology, https://www.mysql.com/products/cluster/, November 2016.

[6] The Hadoop technologies, http://hadoop.apache.org/, November 2016.

5.2 Pre-Processing Module

The pre-processing module determines and filters network data in four steps. Firstly, its feature conversion replaces symbolic features with numeric ones because our DMM-based ADS technique deals with only numeric attributes. Secondly, its feature reduction uses the PCA technique to adopt a small number of uncorrelated features. As the PCA technique is one of the best-known linear feature reduction techniques due to its the advantages. It requires less memory storage, having lower data transfer and processing times, as well as better detection accuracy than others [22, 26]. So, we chose it for this study.

Thirdly, feature normalisation arranges the value of each feature in a specific interval to eliminate any bias from raw data and easily visualise and process it. We applied the z-score function, which scales each feature (x) with a 0 mean (μ) and 1 standard deviation (δ), as shown in Fig. 8, to normalise the data using the formula

$$z = \frac{(x - \mu)}{\delta} \tag{6}$$

Another essential statistical measure is the normality test which is a way of assessing whether particular data follow a normal distribution. We used the Kolmogorov-Smirnov (K-S) test, which is one of the most popular, in our previous work [34]. In it, if the data do not follow a normal distribution, mixture models, such as the GMM, BMM and DMM, are used to efficiently define outliers. In this chapter, we use Q-Q plots to show that the network data do not follow a Gaussian distribution. A Q-Q plot is a graphical tool designed to draw two sets of

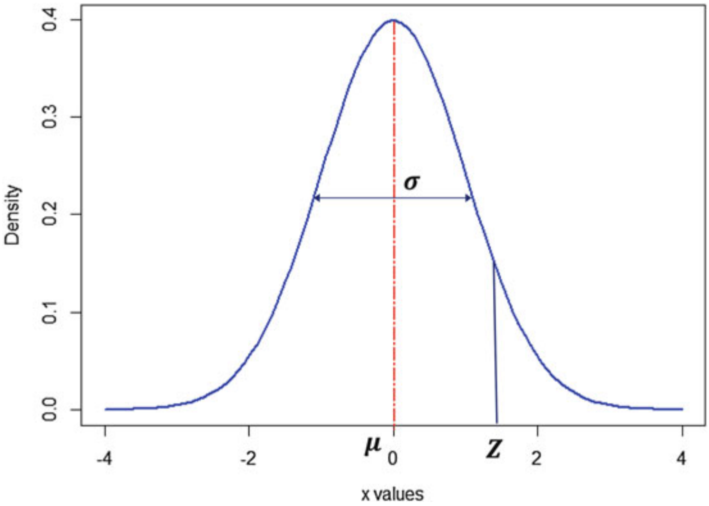

Fig. 8 Gaussian distribution with z-score parameters

quantiles against each other. If these sets are from the same distribution, the points form an almost straight line, with the others treated as outliers [19]. Therefore, it helps to track network flows and define which DE model is the best for identifying suspicious activities as outlier points, as shown in the results in Sect. 6.4. Overall, a statistical analysis is important for network data to make a decision regarding detecting and preventing malicious events.

6 Experimental Results and Discussions

This section discusses the datasets used for evaluating the proposed technique, and then the evaluation metrics applied for assessing the performance of the proposed technique compared with some peer techniques. Finally, the features selected from the NSL-KDD and UNSW-NB15 datasets, with the statistical results of these features are explained.

6.1 Datasets Used for Evaluation

Despite the NSL-KDD and KDD CUP 99 datasets being outdated and having several problems, in particular duplications of records and unbalancing of normal and attack records [33, 47, 48], they are widely used to evaluate NIDSs, due to a lack of accurate dataset availability. As most state-of-the-art detection techniques have been applied to these datasets, which are ultimately from the same network traffic, in order to provide a fair and reasonable evaluation of the performance of our proposed DMM-based ADS technique and comparison with those of related state-of-the-art detection approaches, we adopted the NSL-KDD dataset and contemporary UNSW-NB15 dataset which was recently released.

The NSL-KDD dataset is an improved version of the KDD CUP 99 dataset suggested by Tavallaee et al. in [48]. It addresses some of the problems in the KDD CUP 99 dataset, such as removing redundant records in the training and testing sets to eliminate any classifier being biased towards the most repeated records. Like in this dataset, in the NSL-KDD dataset, each record has 41 features and the class label. It consists of five different classes, one normal and four attack types (i.e., DoS, Probe, U2R and R2L), and includes two sets, training (‘$KDDTrain^{+} - FULL$’ and ‘$KDDTrain^{+} - 20\%$’) and testing (‘$KDDTest^{+} - 20\%$’ and ‘$KDDTest^{21} - newattacks$’).

The UNSW-NB15 dataset has a hybrid of authentic contemporary normal and attack records. The volume of its network packets is approximately 100 GB with 2,540,044 observations logged in four CSV files. Each observation has 47 features and the class label which demonstrate its variety in terms of high dimensionality. Its velocity is, on average, 5–10 MB/s between sources and destinations which means higher data rate transmissions across the Ethernets which exactly mimic real network environments. The UNSW-NB15 dataset includes ten different classes, one normal and nine security and malware types (i.e., Analysis, Backdoors, DoS, Exploits, Generic, Fuzzers for anomalous behaviours, Reconnaissance, Shellcode and Worms) [33, 34].

6.2 Performance Evaluation

Several experiments were conducted on the two datasets to measure the performance and effectiveness of the proposed DMM-based ADS technique using external evaluation metrics, including the accuracy, DR and FPR which depend on the four terms true positive (TP), true negative (TN), false negative (FN) and false positive (FP). TP is the number of actual attack records classified as attacks, TN is the number of actual normal records classified as normal, FN is the number of actual attack records classified as normal and FP is the number of actual normal records classified as attacks. These metrics are defined as follows.

- The **accuracy** is the percentage of all normal and attack records correctly classified, that is,

$$accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{7}$$

- The **Detection Rate (DR)** is the percentage of correctly detected attack records, that is,

$$DR = \frac{TP}{(TP + FN)} \tag{8}$$

- The **False Positive Rate (FPR)** is the percentage of incorrectly detected attack records, that is,

$$FPR = \frac{FP}{(FP + TN)} \tag{9}$$

6.3 Pre-Processing Phase and Description

The DMM-based ADS technique was evaluated using the eight features from the NSL-KDD and UNSW-NB15 datasets adopted using the PCA listed in Table 2.

The proposed DMM-ADS technique was developed using the ‘R language’ on Linux Ubuntu 14.04 with 16 GB RAM and an i7 CPU processor. To conduct the experiments on each dataset, we selected random samples from the ‘full’ NSL-KDD dataset and the CSV files of the UNSW-NB15 dataset with various sample sizes

Table 2 Attributes selected from two datasets

Selected attributes	Description
NSL-KDD dataset	
srv_count	Number of connections to the same service as the current connection in the past 2 s
dst_host_srv_count	Number of connections to the same service in the past 100 connections
count	Number of connections to the same host as the current connection in the past 2 s
dst_host_same_srv rate	Number of connections to different service as the current connection in the past 2 s
dst_host_count	Number of connections to the same host in the past 100 connections
hot	Hot indicators, e.g., access to system directories, creation, and execution of programs
srv_diff_host_rate	Percentage of same service connections to different hosts
rerror_rate	Percentage of same host connections that have "REJ" errors
UNSW-NB15 dataset	
ct_dst_sport_ltm	Number of connections containing the same destination address and source port in 100 connections
tcprtt	Round-trip time of TCP connection setup computed by the sum of 'synack' and 'ackdat'
dwin	Value of destination TCP window advertisement
ct_src_dport_ltm	Number of connections containing the same source address and destination port in 100 connections
ct_dst_src_ltm	Number of connections containing the same source and destination address in 100 connections
ct_dst_ltm	Number of connections containing the same destination address in 100 connections
smean	Mean of flow packet sizes transmitted from source
service	Service types, e.g., HTTP, FTP, SMTP, SSH, DNS and IRC

between 80,000 and 200,000. In each sample, normal instances were approximately 60–70% of the total size, with some used to create the normal profile and the rest for the testing set.

6.4 Statistical Analysis and Decision Support

Statistical analysis supports the decisions of defining the type of modelling, which efficiently fits data to recognise outliers as attacks. As previously mentioned, the Q-Q plot is a graphical tool to check if a set of data come from a normal theoretical distribution. features are considered from a normal distribution if the values of those features fall on the same theoretical distribution line. Figure 9 represents that the

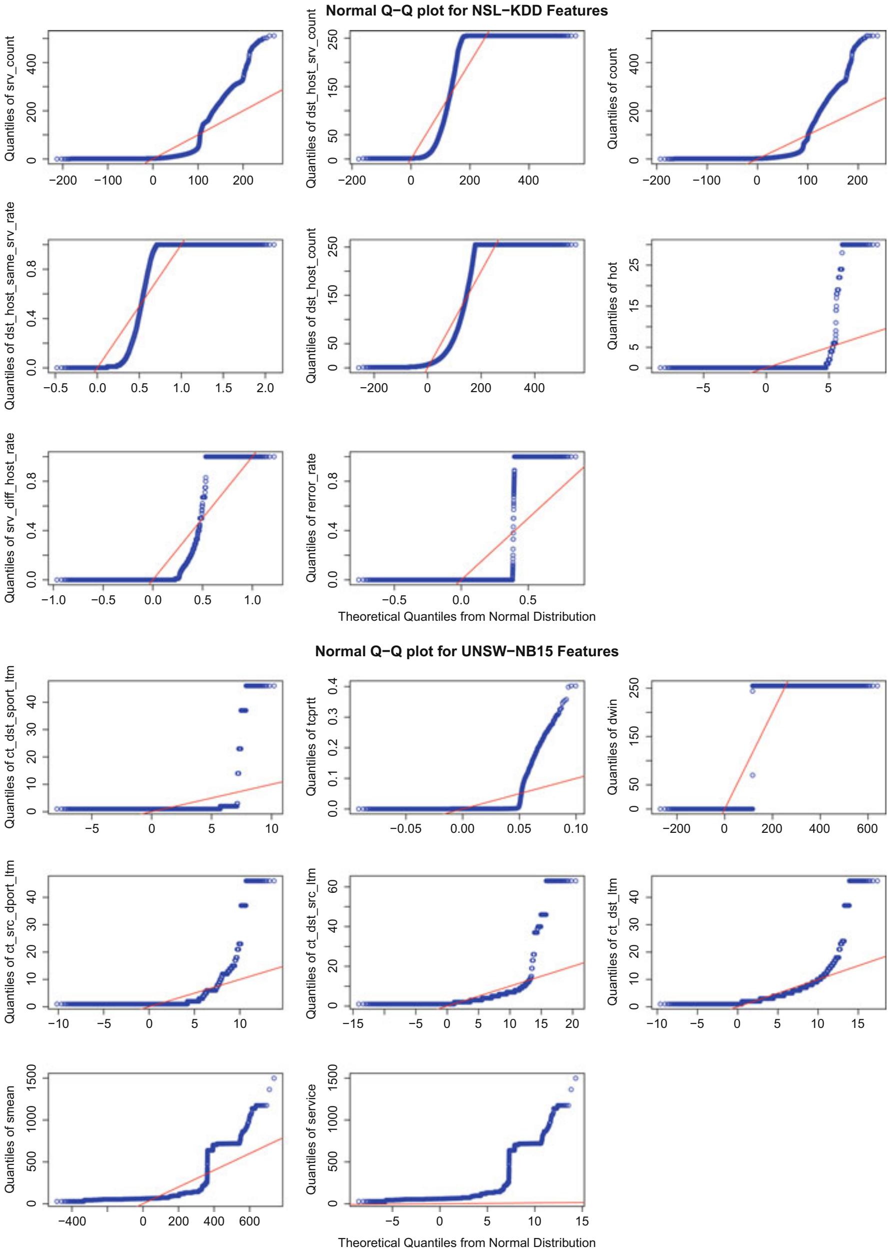

Fig. 9 Q-Q plot of the features selected from both datasets

selected features do not fall on the theoretical distribution lines (i.e., red ones), and there are much greater variations than the lines of feature values. We, therefore, decided to choose the DMM, as one of the best non-normal distribution, for fitting these features to build an ADS based on detecting the too far points of the feature values as anomalies.

The PDFs of the DMM are estimated for normal and abnormal instances using the NSL-KDD and UNSW-NB15 datasets to demonstrate to what extent these instances vary, as presented in Fig. 10. In the NSL-KDD dataset, the PDFs of the normal values range between 0 and 0.20, and their values are between −50 and 0. In contrast, the PDFs of the abnormal values falls between 0 and 0.5, and their values are between −30 and 0. As a result, it is noted that the PDFs of the normal instances are different from the attack ones. Likewise, in the UNSW-NB15 dataset, the PDFs of the normal instances are also dissimilar to the attack instances. These results assert that the proposed decision-making method in Algorithm 2 can effectively detect attacks due to the differences in their PDFs.

6.5 Performance of DMM-Based ADS Technique

The performance evaluation of the DMM-based ADS technique was conducted on the features selected from the two datasets, with the overall DR, accuracy and FPR values listed in Table 3. Figure 11 represents the Receiver Operating Characteristics (ROC) curves which display the relationship between the DRs and FPRs using the w values. It can be seen that the steady increase in the w value between 1.5 and 3 increased the overall DR and accuracy while decreasing the overall FPR.

In the NSL-KDD dataset, when the w value surged steadily from 1.5 to 3, the overall DR and accuracy increased from 93.1% to 97.8% and 93.2% to 97.8%, respectively, while the overall FPR reduced from 3.1% to 2.5%. Likewise, in the UNSW-NB15 dataset, the overall DR and accuracy increased from 84.1% to 93.9% and 89.1% and 94.3%, respectively, but the overall FPR reduced from 9.2% to 5.8% when the w value increased from 1.5 to 3.

Tables 4 and 5 show comparisons of the DRs of the record types for the w values on the NSL-KDD and UNSW-N15 datasets, respectively, which refers that, when the w value increased, the DR gradually improved. It is clear in Table 4 that the DMM-based ADS technique could detect the majority of record types of the NSL-KDD dataset with a normal DR varying between 96.7% and 99.8%, and the lowest FN rate when the w value changed from 1.5 to 3. Similarly, the DRs of the attack types increased gradually from an average of 93.2% to an average of 97.1%.

Table 5 indicates that the DMM-based ADS technique detected record types of the UNSW-NB15 dataset with normal DRs varying from 83.4% to 94.2% when the w value increased from 1.5 to 3. Similarly, the DRs of the attack types increased gradually from an average of 77.5% to an average of 93.2%.

The Shellcode, Fuzzers, Reconnaissance, and Backdoor attacks do not achieve the best DRs with the highest w, whereas the DRs of the other attacks, DoS, Generic,

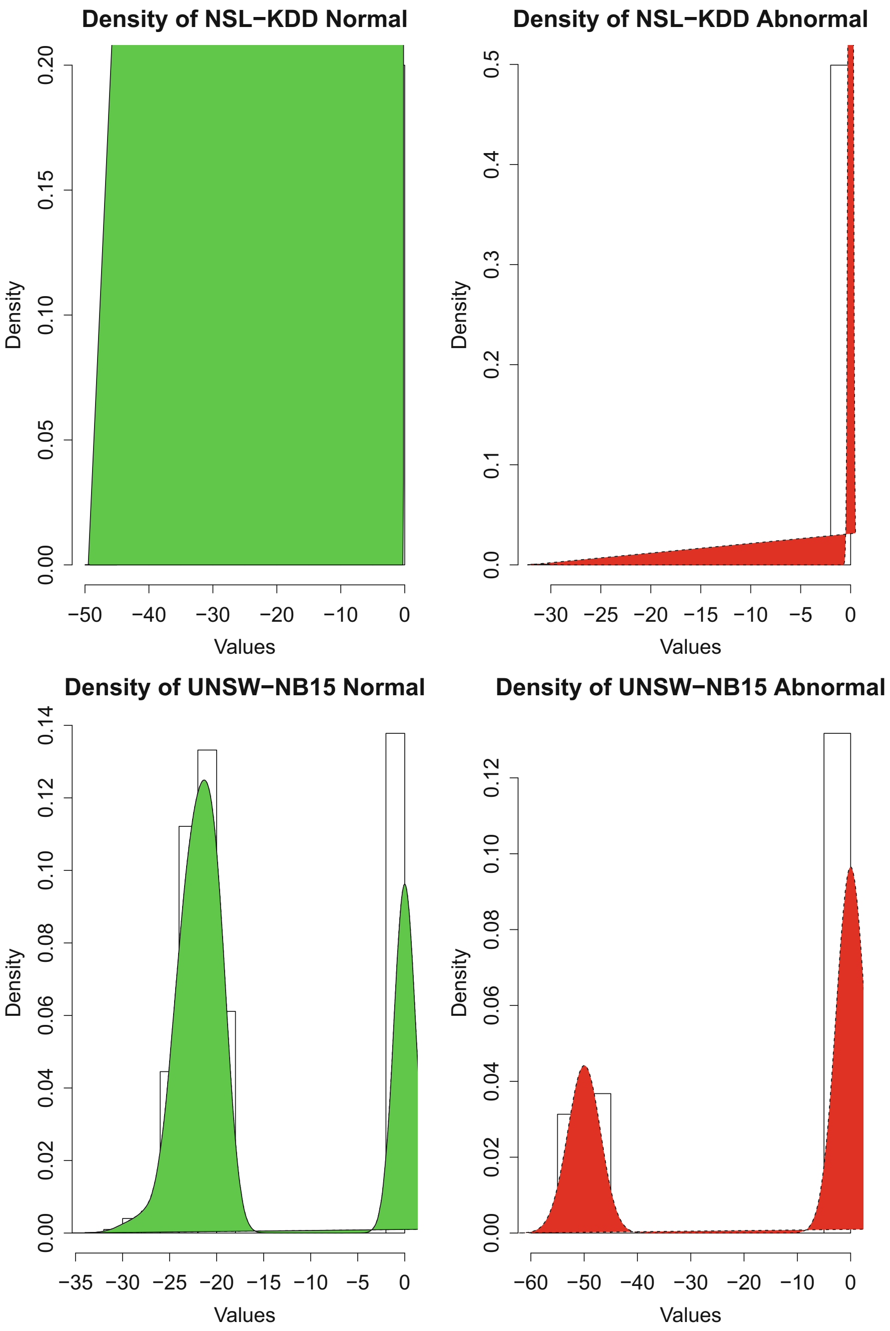

Fig. 10 Normal and abnormal PDFs from a sample of both datasets

Table 3 Performance evaluation of features selected from both datasets

	Datasets					
	NSL-KDD			UNSW-NB15		
w value	Detection rate	Accuracy	False positive rate	Detection rate	Accuracy	False positive rate
1.5	93.1	93.2	3.1	84.1	89.0	9.2
2	93.2	93.8	4.2	87.1	88.1	6.6
2.5	97.3	97.8	2.8	89.2	90.7	7.0
3	97.8	97. 8	2.5	93.9	94.3	5.8

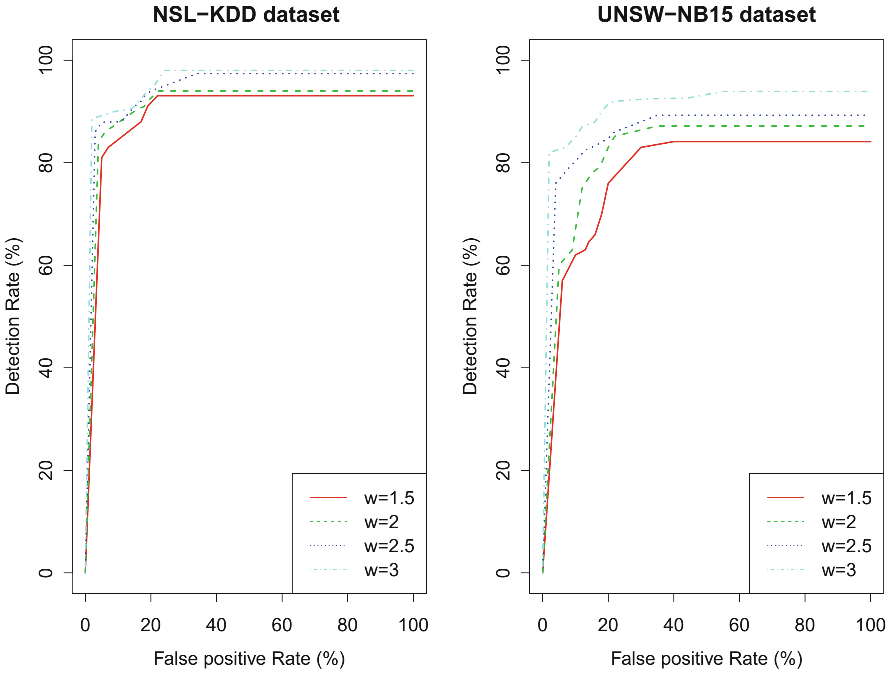

Fig. 11 ROC curves of two datasets with different w values

Exploits, and Worms, are lower, due to the slight similarities between these attack instances and normal ones. It can be noted that, as the variances of the selected features for these instances are close, the PDFs fell into each other in terms of decision-making.

Table 4 Comparison of detection rate (%) on NSL-KDD dataset

Instance type	w values			
	1.5	2	2.5	3
Normal	96.7	97.2	97.3	99.8
DoS	97.0	98.0	98.8	99.7
Probe	92.6	93.7	95.3	97.8
R2L	95.1	93.1	95.1	95.8
U2R	92.6	90.8	93.2	94.0

Table 5 Comparison of detection rate (%) on UNSW-NB15 dataset

Instance type	w values			
	1.5	2	2.5	3
Normal	83.4	83.0	89.7	94.2
DoS	89.1	89.1	88.2	88.1
Backdoor	63.5	72.2	74.2	71.3
Exploits	42.3	78.2	82.1	81.0
Analysis	73.8	76.3	78.0	81.1
Generic	78.1	89.4	88.1	87.7
Fuzzers	43.1	49.1	50.8	52.8
Shellcode	42.2	51.1	52.2	52.2
Reconnaissance	56.1	54.1	57.1	57.2
Worms	37.4	45.2	44.3	48.3

Table 6 Comparison of performances of four techniques

Technique	Detection rate (%)	False positive rate (%)
TANN [49]	91.1	9.4
EDM [46]	94.2	7.2
MCA [47]	96.2	4.9
DMM-based ADS	97.2	2.4

6.6 Comparative Study

The performance evaluation results for the DMM-based ADS technique based on the NSL-KDD dataset were compared with those from other three existing techniques, namely the Triangle Area Nearest Neighbours (TANN) [49], Euclidean Distance Map (EDM) [46] and Multivariate Correlation Analysis (MCA) [47], with their overall DRs and FPRs listed in Table 6. These techniques are used for comparing with our technique because they are the recent ones which have similar statistical measures to our DMM-based ADS. The DRs of The TANN, EDM and MCA were 91.1%, 94.2% and 96.2%, respectively, and their FARs 9.4%, 7.2% and 4.9%, respectively. In contrast, the DMM-based ADS achieved better results of a 97.2% DR and 2.4% FPR.

The key reason for the DMM-based ADS technique performing better than the other techniques was that the DMM fits the boundaries of each feature perfectly, because it has a set of distributions for computing the PDF of each instance. Moreover, the lower-upper IQR method could effectively specify the boundary between normal and outlier instances. However, despite the DMM-based ADS technique achieving the highest DR and lowest FPR on the NSL-KDD dataset, its performance on the UNSW-NB15 dataset was relatively lower due to slight variations between the normal and abnormal instances. This indicated the complicated patterns of contemporary attacks that almost mimic normal patterns.

6.7 Advantages and Disadvantages of DMM-Based ADS

The DMM-based ADS has several advantages. To start with, it is easily deployed on large-scale systems to detect malicious activity in real-time because its training and testing phases depend only on the DMM parameters of the normal profile. Since the decision-making method is used the lower-upper IQR rule as a threshold, it can identify the class label of each record with no dependency on other records. Moreover, the ease of updating the normal profile parameters, with respect to choose the best threshold. In contrast, if there are higher similarities between features, it will produce higher FPR, so we applied the PCA to reduce the number of features with selecting the highest variation of features for improving the performance of the proposed technique. Also, the DMM-based ADS cannot define attack types, such DoS and backdoors, as it was designed for handling binary classification (i.e., normal or attacks). For addressing this limitation, we will design a new statistical function to identify the PDF values of each attack type.

7 Conclusion

This chapter discussed a proposed scalable framework consisting of three main modules, namely, capturing and logging, pre-processing and a statistical decision engine. The purpose of the first module was to sniff and collect network data from a distributed database to easily handle large-scale environments while the second analysed and filtered network data to improve the performance of the decision engine. Finally, the third, the Dirichlet mixture model-based anomaly detection, was designed based on an anomaly detection methodology for recognising abnormal data using a lower-upper interquartile range as a decision-making method. The empirical results showed that a statistical analysis, such as Q-Q plots, helped to make a decision regarding choosing the best model for identifying attacks as outliers. The performance evaluation of the Dirichlet mixture model-based anomaly detection demonstrated that it was more accurate than some other significant methods. In future, we will investigate other statistical methods, such as a particle filter, with the aim of integrating them with the Q-Q plots to design a visual application for analysing and monitoring network data, and making decisions regarding specific intrusions. We will also extend this study to apply the architecture of the proposed framework in cloud computing and SCADA systems.

References

1. Aburomman, A.A., Reaz, M.B.I.: A novel svm-knn-pso ensemble method for intrusion detection system. Applied Soft Computing **38**, 360–372 (2016)
2. Ahmed, M., Mahmood, A.N., Hu, J.: A survey of network anomaly detection techniques. Journal of Network and Computer Applications **60**, 19–31 (2016)
3. Alqahtani, S.M., Al Balushi, M., John, R.: An intelligent intrusion detection system for cloud computing (sidscc). In: Computational Science and Computational Intelligence (CSCI), 2014 International Conference on, vol. 2, pp. 135–141. IEEE (2014)
4. Ambusaidi, M., He, X., Nanda, P., Tan, Z.: Building an intrusion detection system using a filter-based feature selection algorithm (2016)
5. traffic analysis, N.: Network traffic analysis (November 2016). URL https://www.ipswitch.com/solutions/network-traffic-analysis
6. Berthier, R., Sanders, W.H., Khurana, H.: Intrusion detection for advanced metering infrastructures: Requirements and architectural directions. In: Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on, pp. 350–355. IEEE (2010)
7. Bhuyan, M.H., Bhattacharyya, D.K., Kalita, J.K.: Network anomaly detection: methods, systems and tools. IEEE Communications Surveys & Tutorials **16**(1), 303–336 (2014)
8. Bouguila, N., Ziou, D., Vaillancourt, J.: Unsupervised learning of a finite mixture model based on the dirichlet distribution and its application. IEEE Transactions on Image Processing **13**(11), 1533–1543 (2004)
9. Boutemedjet, S., Bouguila, N., Ziou, D.: A hybrid feature extraction selection approach for high-dimensional non-gaussian data clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**(8), 1429–1443 (2009)
10. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM computing surveys (CSUR) **41**(3), 15 (2009)
11. Corona, I., Giacinto, G., Roli, F.: Adversarial attacks against intrusion detection systems: Taxonomy, solutions and open issues. Information Sciences **239**, 201–225 (2013)
12. Ding, Q., Kolaczyk, E.D.: A compressed pca subspace method for anomaly detection in high-dimensional data. IEEE Transactions on Information Theory **59**(11), 7419–7433 (2013)
13. Dua, S., Du, X.: Data mining and machine learning in cybersecurity. CRC press (2016)
14. Dubey, S., Dubey, J.: Kbb: A hybrid method for intrusion detection. In: Computer, Communication and Control (IC4), 2015 International Conference on, pp. 1–6. IEEE (2015)
15. Escobar, M.D., West, M.: Bayesian density estimation and inference using mixtures. Journal of the american statistical association **90**(430), 577–588 (1995)
16. Fahad, A., Tari, Z., Almalawi, A., Goscinski, A., Khalil, I., Mahmood, A.: Ppfscada: Privacy preserving framework for scada data publishing. Future Generation Computer Systems **37**, 496–511 (2014)
17. Fan, W., Bouguila, N., Ziou, D.: Unsupervised anomaly intrusion detection via localized bayesian feature selection. In: 2011 IEEE 11th International Conference on Data Mining, pp. 1032–1037. IEEE (2011)
18. Fan, W., Bouguila, N., Ziou, D.: Variational learning for finite dirichlet mixture models and applications. IEEE transactions on neural networks and learning systems **23**(5), 762–774 (2012)

19. Ghasemi, A., Zahediasl, S., et al.: Normality tests for statistical analysis: a guide for non-statisticians. International journal of endocrinology and metabolism **10**(2), 486–489 (2012)
20. Giannetsos, T., Dimitriou, T.: Spy-sense: spyware tool for executing stealthy exploits against sensor networks. In: Proceedings of the 2nd ACM workshop on Hot topics on wireless network security and privacy, pp. 7–12. ACM (2013)
21. Greggio, N.: Learning anomalies in idss by means of multivariate finite mixture models. In: Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference on, pp. 251–258. IEEE (2013)
22. Harrou, F., Kadri, F., Chaabane, S., Tahon, C., Sun, Y.: Improved principal component analysis for anomaly detection: Application to an emergency department. Computers & Industrial Engineering **88**, 63–77 (2015)
23. Horng, S.J., Su, M.Y., Chen, Y.H., Kao, T.W., Chen, R.J., Lai, J.L., Perkasa, C.D.: A novel intrusion detection system based on hierarchical clustering and support vector machines. Expert systems with Applications **38**(1), 306–313 (2011)
24. Hung, S.S., Liu, D.S.M.: A user-oriented ontology-based approach for network intrusion detection. Computer Standards & Interfaces **30**(1), 78–88 (2008)
25. Jadhav, A., Jadhav, A., Jadhav, P., Kulkarni, P.: A novel approach for the design of network intrusion detection system (nids). In: Sensor Network Security Technology and Privacy Communication System (SNS & PCS), 2013 International Conference on, pp. 22–27. IEEE (2013)
26. Lee, Y.J., Yeh, Y.R., Wang, Y.C.F.: Anomaly detection via online oversampling principal component analysis. IEEE Transactions on Knowledge and Data Engineering **25**(7), 1460–1470 (2013)
27. Li, W., Mahadevan, V., Vasconcelos, N.: Anomaly detection and localization in crowded scenes. IEEE transactions on pattern analysis and machine intelligence **36**(1), 18–32 (2014)
28. Milenkoski, A., Vieira, M., Kounev, S., Avritzer, A., Payne, B.D.: Evaluating computer intrusion detection systems: A survey of common practices. ACM Computing Surveys (CSUR) **48**(1), 12 (2015)
29. Minka, T.: Estimating a dirichlet distribution (2000)
30. Modi, C., Patel, D., Borisaniya, B., Patel, H., Patel, A., Rajarajan, M.: A survey of intrusion detection techniques in cloud. Journal of Network and Computer Applications **36**(1), 42–57 (2013)
31. Moustafa, N., Slay, J.: A hybrid feature selection for network intrusion detection systems: Central points. In: the Proceedings of the 16th Australian Information Warfare Conference, Edith Cowan University, Joondalup Campus, Perth, Western Australia, pp. 5–13. Security Research Institute, Edith Cowan University (2015)
32. Moustafa, N., Slay, J.: The significant features of the unsw-nb15 and the kdd99 data sets for network intrusion detection systems. In: Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS), 2015 4th International Workshop on, pp. 25–31. IEEE (2015)
33. Moustafa, N., Slay, J.: Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In: Military Communications and Information Systems Conference (MilCIS), 2015, pp. 1–6. IEEE (2015)
34. Moustafa, N., Slay, J.: The evaluation of network anomaly detection systems: Statistical analysis of the unsw-nb15 data set and the comparison with the kdd99 data set. Information Security Journal: A Global Perspective (2016)
35. Nadiammai, G., Hemalatha, M.: An evaluation of clustering technique over intrusion detection system. In: Proceedings of the International Conference on Advances in Computing, Communications and Informatics, pp. 1054–1060. ACM (2012)
36. Naldurg, P., Sen, K., Thati, P.: A temporal logic based framework for intrusion detection. In: International Conference on Formal Techniques for Networked and Distributed Systems, pp. 359–376. Springer (2004)

37. Perdisci, R., Gu, G., Lee, W.: Using an ensemble of one-class svm classifiers to harden payload-based anomaly detection systems. In: Sixth International Conference on Data Mining (ICDM'06), pp. 488–498. IEEE (2006)
38. Pontarelli, S., Bianchi, G., Teofili, S.: Traffic-aware design of a high-speed fpga network intrusion detection system. IEEE Transactions on Computers **62**(11), 2322–2334 (2013)
39. Ranshous, S., Shen, S., Koutra, D., Harenberg, S., Faloutsos, C., Samatova, N.F.: Anomaly detection in dynamic networks: a survey. Wiley Interdisciplinary Reviews: Computational Statistics **7**(3), 223–247 (2015)
40. Rousseeuw, P.J., Hubert, M.: Robust statistics for outlier detection. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **1**(1), 73–79 (2011)
41. Saligrama, V., Chen, Z.: Video anomaly detection based on local statistical aggregates. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pp. 2112–2119. IEEE (2012)
42. Seeberg, V.E., Petrovic, S.: A new classification scheme for anonymization of real data used in ids benchmarking. In: Availability, Reliability and Security, 2007. ARES 2007. The Second International Conference on, pp. 385–390. IEEE (2007)
43. Shameli-Sendi, A., Cheriet, M., Hamou-Lhadj, A.: Taxonomy of intrusion risk assessment and response system. Computers & Security **45**, 1–16 (2014)
44. Sheikhan, M., Jadidi, Z.: Flow-based anomaly detection in high-speed links using modified gsa-optimized neural network. Neural Computing and Applications **24**(3–4), 599–611 (2014)
45. Shifflet, J.: A technique independent fusion model for network intrusion detection. In: Proceedings of the Midstates Conference on Undergraduate Research in Computer Science and Mat hematics, vol. 3, pp. 1–3. Citeseer (2005)
46. Tan, Z., Jamdagni, A., He, X., Nanda, P., Liu, R.P.: Denial-of-service attack detection based on multivariate correlation analysis. In: International Conference on Neural Information Processing, pp. 756–765. Springer (2011)
47. Tan, Z., Jamdagni, A., He, X., Nanda, P., Liu, R.P.: A system for denial-of-service attack detection based on multivariate correlation analysis. IEEE transactions on parallel and distributed systems **25**(2), 447–456 (2014)
48. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A detailed analysis of the kdd cup 99 data set. In: Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications 2009 (2009)
49. Tsai, C.F., Lin, C.Y.: A triangle area based nearest neighbors approach to intrusion detection. Pattern recognition **43**(1), 222–229 (2010)
50. Wagle, B.: Multivariate beta distribution and a test for multivariate normality. Journal of the Royal Statistical Society. Series B (Methodological) pp. 511–516 (1968)
51. Wu, S.X., Banzhaf, W.: The use of computational intelligence in intrusion detection systems: A review. Applied Soft Computing **10**(1), 1–35 (2010)
52. Zainaddin, D.A.A., Hanapi, Z.M.: Hybrid of fuzzy clustering neural network over nsl dataset for intrusion detection system. Journal of Computer Science **9**(3), 391 (2013)
53. Zuech, R., Khoshgoftaar, T.M., Wald, R.: Intrusion detection and big heterogeneous data: a survey. Journal of Big Data **2**(1), 1 (2015)