Novel Geometric Area Analysis Technique for Anomaly Detection Using Trapezoidal Area Estimation on Large-Scale Networks

Nour Moustafa, *Student Member, IEEE*, Jill Slay, *Member, IEEE*, and Gideon Creech, *Member, IEEE*

Abstract—The prevalence of interconnected appliances and ubiquitous computing face serious threats from the hostile activities of network attackers. Conventional Intrusion Detection Systems (IDSs) are incapable of detecting these intrusive events as their outcomes reflect high false positive rates (FPRs). In this paper, we present a novel Geometric Area Analysis (GAA) technique based on Trapezoidal Area Estimation (TAE) for each observation computed from the parameters of the Beta Mixture Model (BMM) for features and the distances between observations. As this GAA-based detection depends on the methodology of anomaly-based detection (ADS), it constructs the areas of normal observations in a normal profile with those of the testing set estimated from the same parameters to recognise abnormal patterns. We also design a scalable framework for handling large-scale networks, and our GAA technique considers a decision engine module in this framework. The performance of our GAA technique is evaluated using the NSL-KDD and UNSW-NB15 datasets. To reduce the high-dimensional data of network connections, we apply the Principal Component Analysis (PCA) and evaluate its influence on the GAA technique. The empirical results show that our technique achieves a higher detection rate and lower FPR with a lower processing time than other competing methods.

Index Terms—Geometric area analysis, beta mixture model, trapezoidal area estimation, anomaly detection system, large-scale network

1 Introduction

Despite providing network security solutions, existing techniques, including firewall, data encryption and authentication systems that are located on the first line of defence, cannot offer complete protection for computer systems and networks. Cyber attacks are sets of events which attempt to compromise the principles of Confidentiality, Integrity and Availability (CIA) in computer systems [1], [2]. Each type has its own sophisticated style that imposes a serious issue for detection, for example, a DoS attack corrupts computer resources which breaks the availability principle, while malware codes hijack the execution flow of applications which violates the integrity principle [3]. Existing security mechanisms have a difficulty in detecting stealth and zero-day attacks [4], an additional line of defence against these attacks is designed, for instance, anti-virus software and Intrusion Detection Systems (IDSs), have been considered. Therefore, a complete layered defence strategy is required by integrating the two lines to currently afford a much more comprehensive defence.

With the aim of obtaining a solid line of defence against low-footprint attacks, a plethora of research has been undertaken to design an intelligent IDS which are classified into two major categories, namely, Misuse-based (MDS) and Anomaly-based (ADS) [5], [6], [7]. On the one hand, a MDS monitors network traffic or host traces to match observed behaviours against a known blacklist. However, although it provides higher detection rates and lower FPRs, it cannot define any zero-day attacks or even variants of existing attacks. Moreover, it requires a huge effort to frequently update the blacklist which is a set of rules for each attack type generated by security expertise [8]. On the other hand, an ADS creates a legitimate profile of network or host events and detects any deviation from it as an anomaly. As it can detect both existing and new attacks, such as zero-day attacks and, unlike MDS, does not require any effort to generate rules, it is a better solution than a MDS if its detection method is correctly designed [3], [8], [9], [10].

Nevertheless, an ADS still faces two major problems in terms of the use of its architecture in industry for large-scale networks [7], [11], [12]. First, the creation of a comprehensive profile from diverse normal patterns is very difficult to be achieved. Second, the methodology for building an adaptive and lightweight detection method which efficiently discriminates between legitimate and suspicious activity on high speed and large network environments. Since networks contain different interconnected appliances, software and platforms to provide users and organisations with services anytime, anywhere, analysing those which have flows with large volumes, high-velocity transmissions and include a wide variety of high dimensionality, is difficult. ADSs have been developed based on the techniques of data mining and machine learning [5], [13], artificial intelligence

- *The authors are with the Australian Centre for Cyber Security, University of New South Wales, Canberra Campus, Northcott Dr, Canberra, ACT 2600, Australia.*
  E-mail: nour.moustafa@unsw.edu.au, {j.slay, G.Creech}@adfa.edu.au.

Manuscript received 24 Aug. 2016; revised 25 Apr. 2017; accepted 10 June 2017. Date of publication 14 June 2017; date of current version 27 Nov. 2019. (Corresponding author: Nour Moustafa.)
Recommended for acceptance by K.-K. R. Choo, M. Conti, and A. Dehghantanha.
Digital Object Identifier no. 10.1109/TBDATA.2017.2715166

[14], knowledge-based [15] and statistical [8], [16] models. However, their proposed mechanisms predominantly reflect high false positive rates due to the difficulty of finding a solution that mitigates the aforementioned challenges. Recent studies [7], [8], [11], [17], [18], [19] have concentrated on statistical analysis because of the ease of simultaneously implementing and determining intrinsic potential characteristics of legitimate network patterns for both attributes/features and observations/records. However, ascertaining the potential characteristics and specifying a certain threshold for a detection method demands accurate analysis.

In this study, for the first time in this field, we propose a novel Geometric Area Analysis Technique (GAA) for an ADS. The core theories behind it come from computing the Trapezoidal Area Estimation (TAE) for each record from the estimated parameters of the Beta Mixture Model (BMM) for features and the distances between records. In the literature, each approach is used separately or integrated with other methods to achieve other functions. The BMM generally estimates the prior of Bayesian inference [20], while the TAE is used as a membership function in the fuzzy logic technique [21]. In [22], [23], the features of network traffic did not follow a Gaussian distribution, a beta distribution could fit them efficiently with a lower noise [20], [24], [25]. Also as, in [26], [27], [28], the TAE provided an accurate area for a random variable.

Therefore, we combine the BMM and distances between records in order to compute the TAE for record areas which are then used to discriminate between those that are normal and abnormal by a proposed decision method. We also suggest a scalable framework to effectively handle large-scale networks. This framework comprises of a data sniffing and storing module to capture and log network data, data preprocessing module to determine and filter these data, and our GAA technique to identify malicious behaviours. The details of the GAA technique and the framework are provided in Sections 3 and 4, respectively.

The key contributions of this paper are as follows.

- We propose a new GAA technique based on the TAE that computes the areas for records using the BMM and the distances between records from which it establishes a legitimate profile in the training phase and computes the area for each testing record. To detect attack records, we develop a new decision method that checks each testing record' s area and, if it falls into the legitimate profile, it will consider as a normal record, otherwise an attack one.
- An in-depth mathematical analysis of our GAA technique is presented to help the use of it in other domains. Furthermore, its computational complexity and processing time are provided to demonstrate that it can easily be designed for online processing.
- A performance evaluation of this framework is conducted on two NIDS datasets: the NSL-KDD, which is an enhanced version of the KDD99 [29], [30]; and UNSW-NB15 [2], [31]. Although the KDD dataset is outdated and does not contain contemporary patterns of attacks, it is still publicly used to evaluate NIDSs. Then, to recognise modern attacks, we use the UNSW-NB15 dataset.

The rest of this paper is organised as follows. Background and previous work related to ADS and our novel GAA technique are explained in Section 2. Section 3 discusses details of the novel GAA technique, and Section 4 explains the architectural framework for establishing a scalable, adaptive, and lightweight ADS. Descriptions of the benchmark NSL-KDD and UNSW-NB15 datasets are provided in Section 5. Section 6 presents the experimental results and analysis of the proposed GAA-ADS using the two datasets. Finally, we conclude our work and suggest directions for further research.

2 Background and Related Works

An IDS is a mechanism for monitoring and analysing events which occur in a computer system or network to identify possible threats [9], [11], [32]. In this paper, we focus on the ADS methodology because it can detect both known and unknown attacks. A typical ADS consists of three modules: a data source; data pre-processing; and decision engine. The data source includes raw audit data collected from network traffic or host traces while data pre-processing involves the creation of features from audit data which are then passed to the decision engine that is used to discriminate between legitimate and suspicious activities and is considered the major module [5], [6], [7], [9], [11].

Over the last decade, an ADS has been proposed for mitigating security intrusions. However, with the high speed and size of current networks, this methodology still faces the challenge of building a scalable, adaptable, and lightweight ADS [11], [12]. Statistical approaches for establishing such a system have been recommended in many studies [7], [8], [11], [17], [19]. For instance, Robert et al. [18] developed a probabilistic detection model using Stochastic Petri-Nets and a beta distribution to detect abnormal behaviours in a cyber-physical system. The model recognised the strengths of detection and response in terms of the detection interval, number of detectors and per-host minimum compliance threshold.

A Bayesian inference approach which developed a collaboration framework for data networks by accumulating feedbacks from distributed sensors was proposed in [33] and integrated with a beta distribution to model the false and true positive rates for each IDS. Zhiyuan et al. [8] proposed a Multivariate Correlation Analysis for building a DoS attack detection model, with a triangular area-based technique established to capture the correlation between features to assist in detecting suspicious behaviours. Also, in [17], a geometrical distance analysis algorithm for building an effective NIDS based on mining the correlations between the attributes of payload packets was provided. Kamaldeep et al. [19] proposed a distributed framework using the random forest technique for detecting peer-to-peer Botnet on a large-scale network. Wentao et al. [34] developed an ADS using an incremental Bayesian inference for clustering large network data. Their technique is based on a mixture of generalised Dirichlet distributions to automatically define legitimate and anomalous clusters.

Although most of the above studies achieved improving in the detection accuracy, as they employed a certain threshold in the detection phase which could be either a binary value (i.e., 0 for normal and 1 for an attack) or constant value that did not reflect from real network traffic; thereby their findings were biased towards legitimate activities that

produced high FPRs [35]. However, if this threshold is computed from the processed data and dynamically changed with the network data, it would reduce the FPR, as presented in this paper.

Some studies have investigated using the TAE as a complementary function for detection purposes, in either network traffic or other fields, such as computer vision, signal processing and neuroscience. For instance, Niandong et al. [26] proposed a fuzzy expert system for network forensics to analyse computer crimes and provide automated digital evidence. Trapezoidal- and rectangular- shaped functions were used as fuzzy membership functions to represent a degree of truth about the number of ping attacks. Likewise, these functions were utilised to build a fuzzy anomaly detection for minimising energy consumption [21].

Nedevschi et al. [28] designed a lane stereovision detection model using a trapezoidal rule for lane model matching. In [27], Yuan et al. proposed a detection method based on trapezoidal rules for defining the windshield regions of vehicles which defined each region as a trapezoidal area via the colours, shapes and locations of parts of the vehicles. Jongsuebsuk et al. [36] suggested a fuzzy genetic technique to classify malicious network activities. The fuzzy trapezoidal rule was applied to identify attack data, while the genetic technique was used for finding suitable fuzzy rules. However, in this study, we use the TAE for estimating the area of each connection record to efficiently detect malicious activities from the normal profile.

A significant amount of research has been conducted using the PCA to determine the potential characteristics of network traffic, and remove the irrelevant or noisy features, so we use it in the data pre-processing module of the framework to tackle the variety problem of Big Data, as in [37], [38]. The PCA is an approach which selects a small number of uncorrelated features, so-called 'principal components', from a massive number of features since its target determines the highest variance with the lowest number of principal components [39].

An adaptive IDS developed using a hybrid of SVM and PCA was evaluated on the KDD99 dataset, with the results reflecting that the PCA improved both accuracy and processing speed [40]. Zargar et al. [41] suggested category-based intrusion detection based on the PCA to reduce the number of features in the DARPA 1998 dataset. Their empirical results showed that the model provided a smaller set of features that led to an increase in the speed of detection with the same accuracy. More recently, Eduardo et al. [42] proposed a hybrid statistical mechanism using the PCA, the Fisher Discrimination Ratio and Probabilistic Self-Organising Maps (SOMs) to remove unnecessary features when constructing an adaptive IDS.

3 Geometric Area Analysis Technique

The GAA technique calculates the area for each network observation that has a set of features based on its TAE computed from the BMM parameters and distances of records. It creates a profile from normal areas in the training phase, while detecting areas of attack in the decision method.

In the training phase, a normal profile is constructed from legitimate network records by combining BMM estimations and the distances between the mean of normal records and each record. In the testing phase, the estimated parameters of the normal profile are used to compute the area for each observed record. For each of these phases, the TAE is calculated from the output obtained by combining the BMM and distance between records for each individual record, as explained in Section 3.3.

In the decision method, to detect suspicious records, the areas of the normal profile are divided into a number of ordered intervals to reduce the processing time for comparison with the area of each testing record. If the area of a testing record does not belong to the well-known normal intervals, the record will be an attack one.

As decision-making depends on the anomaly–based detection methodology, it is possible to identify an attack record without requiring any prior or relevant information about attack types. Also, The GAA technique does not need a regular update of the attack signature database as is the case for MDS, although it requires regular updates of the normal baseline as is the case for any ADS. However, the high performance of our GAA technique relies on the number of intervals of normal areas that prevents overlapping between the normal and attack areas, as described in Section 3.4.

3.1 Beta Mixture Model

Although a Gaussian mixture model can represent any arbitrary distribution with suitable mixture components, some of these components do not accurately characterise edges while observed data are semi-bounded or bounded [20]. Our study in [5] showed that the features of network traffic cannot efficiently fit a Gaussian distribution as they do not follow its symmetric and unbounded boundary $(]-\infty, \infty[)$. We observe that, in the NSL-KDD and UNSW-NB15 datasets, they are in the semi-bounded $[0, N]$ range, where N is an asymmetric integer or real number. A beta distribution has a more resilient shape than a Gaussian distribution [20], [24], [25] and models random variables that have a finite range ($[a, b], a, b \in \mathbb{R}$), specifically a $[0, 1]$ range, as discussed in Section 3.1.1.

The probability density function (PDF) of a beta distribution is

$$Beta(x; \upsilon, \omega) = \frac{1}{beta(\upsilon, \omega)} x^{\upsilon-\omega}(1-x)^{\omega-1}, \upsilon, \omega > 0 \quad (1)$$

where x is the normalised features, υ and ω indicate the shaped parameters that form the beta distribution, $beta(\upsilon, \omega)$ is the beta function, $beta(\upsilon, \omega) = \Gamma(\upsilon)\Gamma(\omega)/\Gamma(\upsilon + \omega)$ and $\Gamma(.)$ refers to the gamma function $\Gamma(c) = \int_0^\infty \exp(-t)t^{c-1}dt$. If x is a random variable in the beta distribution satisfied Equation (1), its mean μ is

$$\mu = \frac{\upsilon}{(\upsilon + \omega)} \quad (2)$$

The mixture model is a powerful probabilistic method for representing and analysing multivariate data [43]. Since the features of network datasets are multivariate [5], each feature is denoted as a component in the mixture model. In [20], [44], the bounded property data are efficiently modelled by the BMM with less model complexity than the GMM. In our GAA technique, the BMM is used as the first step to compute the PDFs of the network features.

It is perceived that network samples are independent and identically distributed (iid) [5], [8] although multivariate data are, in most cases, statistically dependent. Nevertheless, for any random variable x comprising L elements, the dependency between elements $(x_1, x_2, \ldots, x_L)$ is denoted by a mixture model even if each specific component can only create vectors with statistically independent elements. We define the multivariate BMM for these samples as a PDF of the form

$$\begin{aligned} f(x;\pi,v,\omega) &= \sum_{i=1}^{I} \Pi_i Beta(X, v_i, \omega_i) \\ &= \sum_{i=1}^{I} \Pi_i \prod_{l=1}^{L} Beta(x_l, v_{li}, \omega_{li}) \end{aligned} \tag{3}$$

such that I is the number of mixture components ($X = \{x_1, \ldots, x_L\}$, $\prod = \{\Pi_1, .., \Pi_I\}$, $\upsilon = \{\upsilon_1, \ldots, \upsilon_I\}$, $\omega = \{\omega_1, \ldots, \omega_I\}$), Π_i is the mixing component (where $\sum_{i=1}^{I} \Pi_i = 1, 0 < \pi < 1$), $\{\upsilon_i, \omega_i\}$ are the parameter vectors of the i^{th} mixture component, $Beta(X; \upsilon_i, \omega_i)$ is the component-conditional parameters, and $\{\upsilon_{lI}, ..., \omega_{lI}\}$ are the parameters of the beta distribution for feature x_l.

3.1.1 Parameter Estimation for BMM

The maximum likelihood estimation (MLE) approach [45] is applied to estimate the parameters of the BMM in Equation (3) which fit the observed data. According to its principle, the best parameters ($\theta = \{\upsilon_1, ..., \upsilon_I, \omega_1, ..., \omega_I, \pi_1, ..., \pi_I\}$) maximise the log-likelihood function as

$$\begin{aligned} \mathcal{L}(\theta \mid X) &= \sum_{n}^{N} \log \left[\sum_{i}^{I} \Pi_i Beta(x_n; \upsilon_i, \omega_i) \right] \\ &= \sum_{n}^{N} \log \left[\sum_{i}^{I} \Pi_i \prod_{i=1}^{L} Beta(x_n; \upsilon_i, \omega_i) \right] \end{aligned} \tag{4}$$

The MLE finds the optimal value of θ by considering x_n as the 'incomplete' component-labelled data. Because the latent variables ($z_n = (z_{n1}, ..., z_{nI})^T$) represent an indicator vector that has one element assigned 1, and the remainder ($x_n.x_n$ and z_n) assigned 0, they are used as the 'complete' data, and the likelihood function is reformulated as

$$\begin{aligned} \mathcal{L}(\theta \mid X, Z) &= \sum_{n}^{N} \sum_{i=1}^{I} z_{ni} [\log \Pi_i + \log Beta(x_n; \upsilon_i, \omega_i)] \\ &= \sum_{n}^{N} \sum_{i=1}^{I} z_{ni} \left[\log \Pi_{ni} + \sum_{l=1}^{L} \log Beta(x_{ln}; \upsilon_{li}, \omega_{li}) \right] \end{aligned} \tag{5}$$

In order to iteratively estimate θ, the Expectation-maximisation (EM) algorithm [9] is used. In the E-step, the expected value of z_n is computed as the posterior probability of x_n being established from the i^{th} component which reflects the currently estimated parameters as

$$\bar{z}_{ni} = E[z_{ni}] = \frac{\Pi_i Beta(x_n; \hat{\upsilon}_i, \hat{\omega}_i)}{\sum_{m=1}^{I} \Pi_i Beta(x_n; \hat{\upsilon}_m, \hat{\omega}_m)} \tag{6}$$

In the M-step, the probability (π_i) and parameters ($\theta = \{\upsilon_1, ..., \upsilon_I, \omega_1, ..., \omega_I\}$) are re-estimated given the expected value of the latent variables in order to maximise the log-likelihood, with the updated mixture weight

$$\pi_i = \frac{1}{N} \sum_{n=1}^{N} E[z_{ni}] \tag{7}$$

To compute the two independent parameters ($\hat{\upsilon}_{li}, \hat{\omega}_{li}$), we simultaneously maximise their likelihoods as

$$\begin{aligned} \frac{\partial E[\mathcal{L}_{\mathcal{C}}(\theta \mid X, Z)]}{\partial \upsilon_{li}} &= \sum_{n=1}^{N} \frac{\partial \log Beta(x_{ln}; \upsilon_{li}, \omega_{li})}{\partial \upsilon_{li}} \\ &= \sum_{n=1}^{N} \bar{z}_{ni} \{\log x_{ln} - [\psi(\upsilon_{li}) - \psi(\upsilon_{li} + \omega_{li})]\} \end{aligned} \tag{8}$$

where $\psi(\cdot)$ is the digamma function denoted as

$$\psi(x) = \frac{\partial \log \Gamma(x)}{\partial x} \tag{9}$$

Symmetrically,

$$\begin{aligned} \frac{\partial E[\mathcal{L}_{\mathcal{C}}(\theta \mid X, Z)]}{\partial \omega_{li}} &= \sum_{n=1}^{N} \bar{z}_{ni} \frac{\partial \log Beta(x_{ln}; \upsilon_{li}, \omega_{li})}{\partial \omega_{li}} \\ &= \sum_{n=1}^{N} \bar{z}_{ni} \{\log (1 - x_{ln}) - [\psi(\omega_{li}) - \psi(\upsilon_{li} + \omega_{li})]\} \end{aligned} \tag{10}$$

In order to define the local curvature of $E[\mathcal{L}_{\mathcal{C}}(\theta \mid X, Z)]$, the Hessian matrix ($\mathcal{H}$) is used [44] as

$$\begin{aligned} \mathcal{H}\{E[\mathcal{L}_{\mathcal{C}}(\theta \mid X, Z)]\} &= \begin{bmatrix} \frac{\partial^2 E[\mathcal{L}_{\mathcal{C}}(\theta \mid X, Z)]}{\partial \upsilon_{li} . \partial \upsilon_{li}} & \frac{\partial^2 E[\mathcal{L}_{\mathcal{C}}(\theta \mid X, Z)]}{\partial \upsilon_{li} . \partial \omega_{li}} \\ \frac{\partial^2 E[\mathcal{L}_{\mathcal{C}}(\theta \mid X, Z)]}{\partial \omega_{li} . \partial \upsilon_{li}} & \frac{\partial^2 E[\mathcal{L}_{\mathcal{C}}(\theta \mid X, Z)]}{\partial \omega_{li} . \partial \omega_{li}} \end{bmatrix} \\ &= \begin{bmatrix} \psi'(\upsilon_{li} + \omega_{li}) - \psi'(\upsilon_{li}) & \psi'(\upsilon_{li} + \omega_{li}) \\ \psi'(\upsilon_{li} + \omega_{li}) & \psi'(\upsilon_{li} + \omega_{li}) - \psi'(\omega_{li}) \end{bmatrix} \end{aligned} \tag{11}$$

where $\psi'(x) = d\psi(x)/dx$. Because $\psi'(\upsilon_{li} + \omega_{li}) - \psi'^{(\upsilon_{li})} < 0$, as the indicator $|\mathcal{H}\{E[\mathcal{L}_{\mathcal{C}}(\theta \mid X, Z)]\}| < 0$ shows that the Hessian matrix is a negative definite matrix, it has a local maximum at $\{\hat{\upsilon}_{li}, \hat{\omega}_{li}\}$ which satisfies

$$\begin{bmatrix} \frac{\partial^2 E[\mathcal{L}_{\mathcal{C}}(\theta \mid X, Z)]}{\partial \upsilon_{li}} \\ \frac{\partial^2 E[\mathcal{L}_{\mathcal{C}}(\theta \mid X, Z)]}{\partial \omega_{li}} \end{bmatrix} = 0 \tag{12}$$

From Equation (12), the two updated equations for the parameters $(\upsilon_{li}, \omega_{li})$, which are detailed in [44], can be declared as

$$\begin{aligned} \psi(\hat{\upsilon}_{li}) \; - \; \psi(\hat{\upsilon}_{li} + \hat{\omega}_{li}) &= \frac{\sum_{n=1}^{N} \bar{z}_{ni} \log x_{ln}}{\sum_{n=1}^{N} \bar{z}_{ni}} \\ \psi(\hat{\omega}_{li}) - \psi(\hat{\upsilon}_{li} + \hat{\omega}_{li}) &= \frac{\sum_{n=1}^{N} \bar{z}_{ni} \log (1 - x_{ln})}{\sum_{n=1}^{N} \bar{z}_{ni}} \end{aligned} \tag{13}$$

To describe the modelling of the BMM, given two arbitrary variables (x_1 and x_2), the BMM parameters are computed using the EM algorithm for these variables. In Fig. 1, let the parameters of x_1 (π, v, ω) be (0.55, 30, 10) and those of x_2 (π, v, ω) (0.45, 10, 30). We compute the parameters of the BMM for the features of the network datasets, as listed in Table 2, as the first step in our GAA technique.

3.2 Trapezoidal Area Estimation

The purpose of the GAA technique is to detect the area for each record as either legitimate or suspicious. To achieve

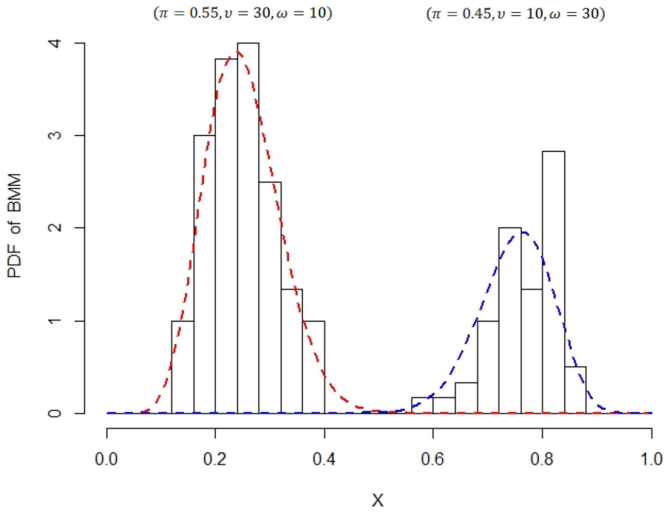

Fig. 1. BMM for two arbitrary variables.

the record by record detection, the TAE estimates the area for each record (r) so that each record has a set of features ($x_{1:D}$, $r_{1:n}$=$\{x_1, x_2, ..., x_D\}$), and tends to be extremely accurate for estimating unequally spaced points. Each PDF of the BMM (i.e., an area under the curve) represents the area of each feature considered as $f(x)$ in the TAE.

The trapezoidal rule is one of the numerical integration families called 'Newton-Cotes formulas' [46]. Its target evaluates $V_n = \int_a^b f(x_{1:d})dx$, where a and b are the lower and upper bounds of each variable (x), the PDF $f(x)$ computed from Equation (1) and D is the number of features generated from the feature reduction discussed in Section 4.2.2.

When a trapezoidal rule is used for multivariate data, it is called a 'composite trapezoidal rule' , as shown in Fig. 2. It is obtained from combining each sub-interval $[x_{d-1}, x_d]$, where $d = 1, 2, ..., D$, as defined in Equation (14). Considering the interval points $a = x_1 < x_2 < \cdots < x_D = b$,

$$\begin{aligned} \int_a^b f(x)dx &= \sum_{i=1}^{d} \int_{x_{d-1}}^{x_d} f(x)dx \\ &\approx \frac{1}{2}\sum_{i=1}^{d}(x_d - x_{d-1})[f(x_{d-1}) + f(x_d)] \end{aligned} \tag{14}$$

In Fig. 2, we assume that the variables from a uniform grid [47] which means that they are of equal length. As a consequence, the total geometric area of each record applied in the normal profile creation and testing phase is

$$\begin{aligned} area(V) &= \int_a^b f(x)dx \\ &= \frac{b-a}{D}[f(x_1) + 2\sum_{i=1}^{D-1} f(x_i) + f(x_D)] \end{aligned} \tag{15}$$

3.3 Normal Profile Creation

The fidelity of constructing a purely normal profile can be achieved by providing secure legitimate traffic which provides credibility of detection. Given a set of normal records ($r_{1:n}^{normal}$) in which each record comprises of a set of features, where $r_{1:n}^{normal} = \{x_1, x_2, ..., x_D\}^{normal}$, the normal profile consists of only statistical observations from $r_{1:n}^{normal}$. These observations include the estimated parameters of the BMM,

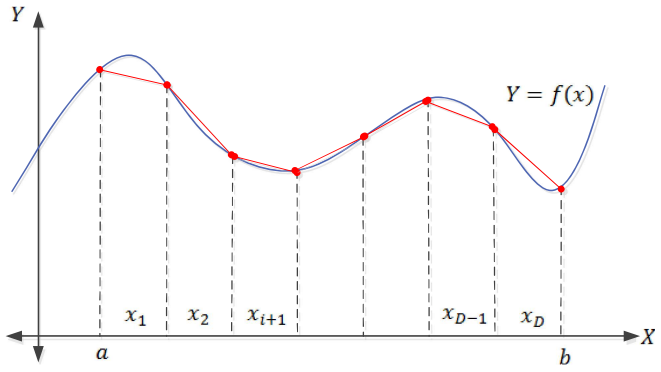

Fig. 2. Composite trapezoidal rule.

the distances between the means of the records using Equation (2). Each distance ($distance_n$) denotes the absolute distance between the mean of all normal records ($\mu = 1/N \sum_{i=1}^{N} \upsilon_i/(\upsilon_i + \omega_i)$) and the mean of each normal record ($\mu_n = 1/D \sum_{d=1}^{D} \upsilon_{nd}/(\upsilon_{nd} + \omega_{nd})$) as

$$distance_n = |\mu - \mu_n| \tag{16}$$

The absolute distance has been successfully applied in cluster and outlier detection mechanisms [48], [49], whereas we use it in our GAA-ADS technique to ensure the detection of dissimilarities between legitimate and suspicious records because, as in [50], distance measures can precisely estimate the dissimilarity between different PDFs. It is practically observed that, if the areas of legitimate records are approximately similar to those of attack records, the sum of each PDF ($f(x_{nD})$) and absolute distance ($distance_n$) clearly distinguishes between them.

Algorithm 1 describes the proposed steps for creating a normal profile ($prof$), with the parameters (π, υ, ω) of the BMM estimated for all the normal records ($r_{1:n}^{normal}$) using Equations (4), (5), (6), (7), (8), (9), (10), (11), (12), and (13) and then used to compute the PDFs of the features ($x_{1:D}$) using Equations (1), (2), (3), (4), and (5). To differentiate between legitimate and attack records, the sum of the PDFs and absolute distances, namely ($filterset_n$), is computed and the TAE of each record ($area_n^{normal}$) calculated by $filterset_n$ using Equation (15).

Algorithm 1. Normal Profile Creation

Input: normal records ($r_{1:n}^{normal}$)
output: normal profile ($prof$)
1: **for** all $r_{1:n}^{normal}$ **do**
2: $[\Pi_d, \upsilon_{nd}, \omega_{nd}] \leftarrow$ estimate the parameters (π, υ, ω) of the BMM using Equations (4), (5), (6), (7), (8), (9), (10), (11), (12), and (13).
3: $f(x_{nd}) \leftarrow$ compute the PDFs using Equations (1), (2), (3), (4), and (5) based on the parameters estimated in Step 2.
4: $\mu_n = 1/D \sum_{d=1}^{D} \upsilon_{nd}/(\upsilon_{nd} + \omega_{nd})$ using Equation (2)
5: **end for**
6: $\mu = 1/N \sum_{i=1}^{N} \upsilon_i/(\upsilon_i + \omega_i)$ using (2)
7: $distance_n = |\mu - \mu_n|$ using (16)
8: $filterset_n \leftarrow (f(x_{nd}) + distance_n)$
9: $area_n^{normal} \leftarrow$ compute the TAE for the $filterset_n$ using (15)
10: sort $area_n^{normal}$ and divide them into K ranges $([min_{Ki}, max_{Ki}])$
11: $prof \leftarrow (\Pi_d, \upsilon_{nd}, \omega_{nd}, \mu_n, [min_{Ki}, max_{Ki}])$
12: **return** $prof$

TABLE 1
Examples of Detecting Attacks Using Estimated Areas

Training phase	0.3	0.35	0.5	0.55	0.73	0.82
Testing phase	normal areas			attack areas		
	0.55	0.79	0.34	0.1	0.72	0.9
Case 1	K=2					
	Ranges label	Min	Max			
	11	0.3	0.5			
	12	0.55	0.82			
decision making						
Record areas	0.55	0.79	0.34	0.1	0.72	0.9
Ranges label	12	12	11	0	12	0
detection	normal	normal	normal	attack	normal(false)	attack
Case 2	k=3					
	Ranges label	min	max			
	21	0.3	0.35			
	22	0.5	0.55			
	23	0.73	0.82			
decision making						
Record areas	0.55	0.79	0.34	0.1	0.72	0.9
Ranges label	22	23	21	0	0	0
detection	normal	normal	normal	attack	attack	attack

As step 10 takes a long processing time for execution because it compares each area of a testing record with all the estimated normal areas ($area_n^{normal}$). These estimated normal areas are sorted and divided into K_i ranges, each K_i represents a minimum and maximum value (min_{Ki} and max_{Ki}, respectively). Mathematically speaking, all the possible values of ranges (i.e., K_{values}) of N observations in a dataset are computed by

$$K_{values} = \{\lfloor N/2 \rfloor, \lfloor (N-1)/2 \rfloor, \lfloor (N-2)/2 \rfloor, \ldots, \lfloor 4/2 \rfloor\}$$
$$Subject\ to,\ K > 1,\ N \geq 4 \tag{17}$$

In Equation (17), the upper and lower values of K are ($\lfloor N/2 \rfloor$) and ($\lfloor 4/2 \rfloor$), respectively, and, as the values following after the lower one are 1 (i.e., $K = 1$), they are excluded as they denote the same original interval; for example, in Table 1, we have $N = 6$ normal areas with their $K_{values} = \{\lfloor 6/2 \rfloor = 3, \lfloor 5/2 \rfloor = 2, \lfloor 4/2 \rfloor = 2\} = \{2, 3\}$, computed using from Equation (17), and these K_{values} are used to generate K_i ranges. These ranges reduce the number of areas of attack falling into normal areas, particularly if normal and attack areas are closed. As a result, normal records can be detected easily at real time if their areas fall into a K_i range, i.e., between min_{Ki} and max_{Ki}, otherwise they are considered attack records, as detailed in Algorithm 2. The estimated parameters ($\Pi, \upsilon_{nd}, \omega_{nd}, \mu_n, [min_{Ki}, max_{Ki}]$) are stored in the normal profile ($prof$) for the testing phase and decision-making method for detecting attacks.

3.4 Testing Phase and Decision-Making Method

In the testing phase, the area ($area_n^{testing}$) of each observed record ($r^{testing}$) is computed using the same estimated parameters of the normal profile ($prof$). Algorithm 2 describes the steps in the testing phase and decision-making method for identifying the areas of attack records. Steps 1 to 4 construct the area of testing records ($area_n^{testing}$) using the stored normal parameters ($\Pi_d, \upsilon_{nd}, \omega_{nd}, \mu_n$).

The decision method is presented in steps 5 to 15 in which each area of a testing record ($area_n^{testing}$) is compared with the area ranges of the normal profile $[min_{Ki}, max_{Ki}]$. If ($area_n^{testing}$) falls within any $[min_{Ki}, max_{Ki}]$ range, it is considered a normal record, otherwise an attack one.

Algorithm 2. Attack Detection Based on Area Estimation

Input: observed record ($r^{testing}$), normal profile ($\Pi_d, \upsilon_{nd}, \omega_{nd}, \mu_n, [min_{Ki}, max_{Ki}]$), flag = 1 // *a record is normal or not*
output: normal or attack record
1: $distance^{testing} = |\mu_n - \mu^{testing}|$
2: compute $f(x_{testing})$ using the parameters $\Pi_d, \upsilon_{nd}, \omega_{nd}$
3: $filterset^{testing} \leftarrow (f(x_{testing}) + distance^{testing})$
4: $area^{testing} \leftarrow$ compute the TAE for the $filterset^{testing}$
5: **for** (i to length (K ranges)) **do**
6: **if** ($area^{testing} \geq min_{Ki}$ && $area^{testing} \leq max_{Ki}$) **then**
7: flag = 0
8: break
9: **end if**
10: **end for**
11: **if** (flag == 0) **then**
12: **return** normal
13: **else**
14: **return** attack
15: **end if**

Table 1 presents an example of detecting attacks based on estimated areas. In the training phase, 6 areas of normal records are assumed while the testing phase contains 3 areas of normal records and 3 of attack ones. The ranges in the training phase are divided using $K = 2$ and $K = 3$ and called 'Case 1' and 'Case 2' , respectively. Case 1 has 2 ranges of (0.3, 0.5) and (0.55, 0.82) labelled '11' and '12' , respectively, and Case 2 has 3 ranges of (0.3, 0.35), (0.5, 0.55) and (0.73, 0.82) labelled '21' , '22' and '23' , respectively. In steps 5 to 11 in Algorithm 2, the detection of attack records indicates that all the records in Case 1 except '0.72' and all those in Case 2 are detected correctly. As a consequence, the K ranges have a significant impact on the detection rate of our GAA technique. With a gradual increase in K from 2 to 3, the DR increases to correctly detect all the normal and attack areas.

4 Framework Architecture

In this section, we propose a scalable framework for the design of an adaptive and lightweight anomaly detection system for efficiently detecting attacks. This framework includes three modules of the data sniffing and storing, data pre-processing, and GAA-ADS technique, as depicted in Fig. 3. In the first module, a set of features is generated from network ingress traffic with existing secure servers used to capture network connections for a well-defined time interval. Monitoring and analysing the traffic at the destination network decreases the overhead of identifying attack records by focusing only on relevant traffic [8].

The second module analyses and filters network traffic in three steps: first, feature conversion replaces symbolic features with numeric ones because our GAA technique deals with only numeric features; second, feature reduction uses the PCA technique to adopt a small number of uncorrelated original features and their principal components; and, third,

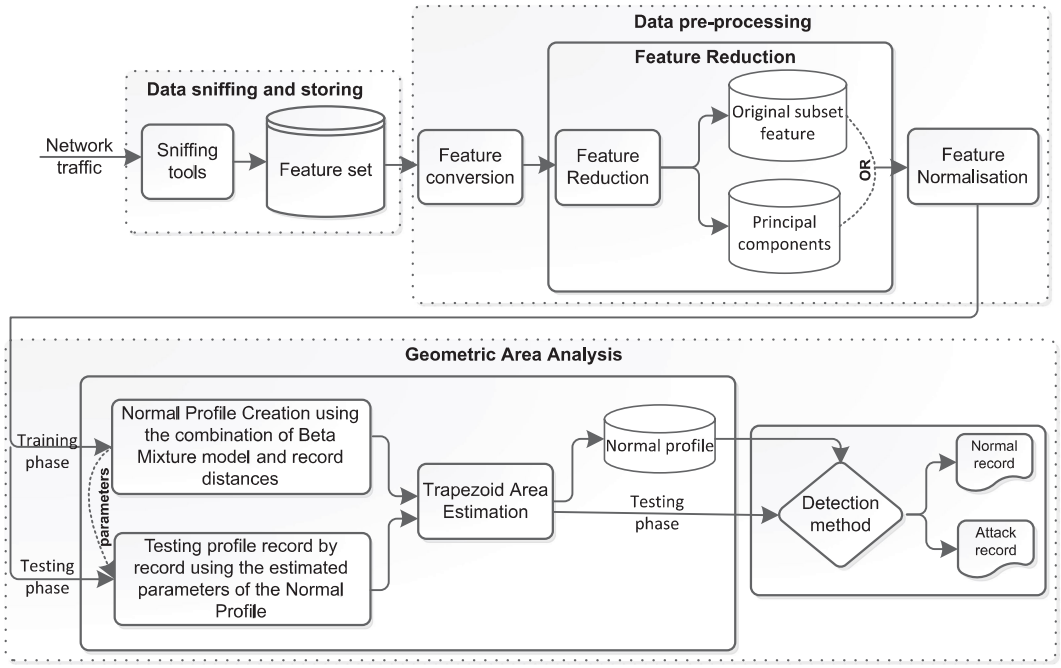

Fig. 3. Proposed framework for building a scalable, adaptive, and lightweight ADS.

feature normalisation, which is an essential step in implementing the BMM, transforms the original subset features or their principal components into a $[0, 1]$ range.

Ultimately, the third module is the new GAA-ADS technique discussed in the previous section which recognises malicious activities based on carefully analysing the geometric areas of normal and attack data. In its training phase, the TAEs of the normal records are computed by combining the BMM estimations and distances of the records. Then, the same parameters are used to compute the TAEs of the testing records while considering any variation from the normal areas an anomaly.

4.1 Data Sniffing and Storing Module

In our previous work [2], we used an IXIA PerfectStorm tool [51] to simulate current realistic normal and attack network traffic, with its testbed configured to simulate a large-scale network. A Tcpdump tool was used for sniffing the packets from the network interface, while Bro, Argus tools and other scripts were applied to mine a set of features, out of the packets. In network systems, it is essential for the creation of features from network packets using ingress devices, such as ingress routers, to aggregate flows. These flows are collected at the destination nodes of each network based on their source/destination IPs and protocols (i.e., flow identifiers) in a particular time window to reduce the overhead of recognising malicious activities [5]. We generated the features of our UNSW-NB15 dataset by designing an extractor module which aggregates packets using the flow identifiers for each of the 100 connection records to ensure relatively easy analysis and classification [2], [31].

These features were logged using the MySQL Cluster CGE technology [52] that has a highly scalable and real-time database for handling online big data. It has a similar architecture to the Hadoop technologies [53] which are the most popular for processing big offline data, but an ADS has to recognise malicious behaviours in a real-time configuration. These generated features are then passed to the data pre-processing module for filtering.

4.2 Data Pre-Processing Module

We use the set of features extracted from network traffic in the NSL-KDD and UNSW-NB15 datasets, which we first process to be compatible input to our GAA technique using the three steps of feature conversion, feature reduction and feature normalisation.

4.2.1 Feature Conversion

Although these datasets have both quantitative and qualitative features, our proposed technique accepts only quantitative ones. Therefore, a unified format for features (X) is applied to convert a symbolic feature into a numeric one (i.e., $X \in \mathbb{R}$, where $\mathbb{R}$ is a real number); for example, the UNSW-NB15 dataset involves three symbolic features, states (e.g., CON, ACC), protocol types (e.g., TCP, UDP) and services (e.g., HTTP, FTP). In each dataset, this function simply replaces the values in the features with ordered numbers, such as CON = 1, ACC = 2, etc.

4.2.2 Feature Reduction

Feature reduction is the process of eliminating irrelevant, redundant or noisy features. In [54], it is divided into feature selection, which finds a subset of the original features, and feature extraction which converts the data from a high-dimensional space to a lower-dimensional space. In [39], [41], [42], [55], the PCA is one of the best-known linear feature reduction mechanisms and has the advantages of requiring less memory storage, data transfer and processing time, as well as increasing detection accuracy compared with other methods [39], [56], [57], [58]. Therefore, after eliciting a set of features from large network data, the first stage

TABLE 2
Features Selected from the Two Datasets

Datasets	Selected features
NSL-KDD	*srv_count, dst_host_srv_count, count, src_bytes, dst_host_same_srv_rate, dst_host_count, srv_diff_host_rate, srv_error_rate, dst_host_srv_error_rate, diff_srv_rate dst_host_rerror_rate, rerror_rate, is_guest_login, num_outbound_cmds, dst_host_srv_diff_host_rate*
UNSW-NB15	*ct_dst_sport_ltm, tcprtt, dwin, sjit, ct_state_ttl ct_src_dport_ltm, dbytes, ct_dst_src_ltm, ct_dst_ltm, smean, dmean, service, proto, dtcpb, ct_src_ltm*

of building a lightweight anomaly detection system is feature reduction in which we apply the PCA due to its advantages. It ranks a set of variables/features based on the highest variance for each feature and creates a new dimensional space of uncorrelated variables by removing low-variance features [39], [56], [59]. The input to our GAA technique is the original features or principal components selected, as presented in Table 2.

4.2.3 Feature Normalisation

An essential step in data pre-processing after reducing the feature set is feature normalisation, which is a method for scaling the value of each attribute into a certain range, with its main benefit removing the bias from raw data without altering their statistical properties. As the first step in the GAA technique is using the beta distribution, which requires a certain range such as $[0, 1]$ for each feature (x_i) as input, the features of a network are normalised into the range of $[0, 1]$ by the linear transformation

$$x_i^{normalised} = (x_i - \min(x)/(\max(x) - \min(x)) \quad (18)$$

5 Description of Benchmark Data Sets

The evaluation of our proposed GAA technique is conducted using the NSL-KDD and UNSW-NB15 datasets, respectively. Despite the NSL-KDD and KDD CUP 99 datasets being outdated and having several problems [2], [30], [60], they are widely used to evaluate NIDSs. Our UNSW-NB15 was recently released to address these problems. Most state-of-the-art detection techniques have been applied to the KDD CUP 99 or NSL-KDD datasets which are ultimately from the same network traffic. Therefore, in order to provide a fair and reasonable comparison of the performances of our proposed GAA-ADS technique and related state-of-the-art detection techniques, we adopt the NSL-KDD and contemporary UNSW-NB15 datasets because they do not contain repeated instances and handle the problem of imbalances between normal and attack observations.

The NSL-KDD dataset is an improved version of the KDD CUP 99 dataset suggested by Tavallaee et al. in [30]. 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 biassed towards the most repeated records. Similarly, 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+_FULL' and 'KDDTest-21_new attacks').

The UNSW-NB15 dataset has a hybrid of authentic contemporary normal and attack records. The volume of its network packets is around 100 Gigabytes generated 2,540,044 observations and are logged in four CSV files. Each observation has 47 features and the class label, and this shows its variety of having high dimensionality. Its velocity is in average 5-10 Megabytes per second between sources and destinations. This means that a higher data rate was transmitted across the Ethernets, which exactly mimics real network environments, as described in [31], [61]. It includes ten different classes, one normal and nine types of security events and malware (i.e., Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Fuzzers for anomalous activity, Shellcode and Worms). A portion of this dataset is used for training and testing sets in [31] and its details are given in [5].

6 Experimental Results and Analysis

6.1 Performance Evaluation

Several experiments are conducted on the two datasets to measure the performance and effectiveness of the proposed GAA-ADS technique using external evaluation metrics, including the accuracy, detection rate 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 the number of actual normal records classified as normal, FN the number of actual attack records classified as normal and FP 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 that are correctly classified, that is,

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (19)$$

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

$$DR = \frac{TP}{(TP + FN)} \quad (20)$$

- The *false positive rate (FPR)* is the percentage of incorrectly detected attack records, that is,

$$FPR = \frac{FP}{(FP + TN)} \quad (21)$$

6.2 Results and Discussion

The GAA-ADS technique is evaluated using 15 original features and their principal components from the NSL-KDD and UNSW-NB15 datasets adopted using the PCA, as presented in Table 2, with more details provided in Appendix 1, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TBDATA.2017.2715166.

TABLE 3
Evaluation of Overall Performances on Original Features

	Datasets					
	NSL-KDD			UNSW-NB15		
K value	DR	Accuracy	FPR	DR	Accuracy	FPR
2	92.1%	92.3%	3.0%	75.1%	77.4%	8.3%
4	92.9%	93.6%	2.6%	81.3%	82.1%	7.4%
6	95.3%	95.5%	0.8%	85.2%	85.7%	7.0%
8	95.4%	95. 6 %	0.7%	89.8%	90.2%	6.9%
10	98.1%	98.8%	0.4%	91.0%	91. 8%	5.8%

TABLE 4
Evaluation of Overall Performances on Principal Components

	Datasets					
	NSL-KDD			UNSW-NB15		
K value	DR	Accuracy	FPR	DR	Accuracy	FPR
2	94.2%	95.0%	1.1%	75.4%	77.6%	8.2%
4	95.1%	95.3%	0.7%	85.2%	86.0%	6.3%
6	96.4%	97.7%	0.2%	87.1%	88.2%	6.1%
8	98.7%	98.8%	0.2%	91.2 %	92.7%	5.9%
10	99.6%	99.7%	0.2%	91.3%	92.8%	5.1%

The proposed GAA-ADS technique is developed using the 'R programming language' on Linux Ubuntu 14.04 with 16 GB RAM and an i7 CPU processor. To conduct the experiments on each dataset, we select random samples from the 'full' NSL-KDD dataset [29] and the CSV files of the UNSW-NB15 dataset with different sample sizes of between 50,000 and 250,000. For each sample size, each normal sample is almost 65-75% of the total size to establish the normal profile (i.e., the training phase), and the others in the testing phase. The performance of the GAA-ADS technique is evaluated on different K values (2, 4, 6, 8 and 10), with a 10-fold cross–validation for the sample sizes to determine their effects. The R code and a complete example using the GAA-ADS technique can be found in Appendices 2 and 3, respectively, available in the online supplemental material.

6.2.1 Performance on Original Features

To provide a better overview of the performance of the GAA-ADS technique on the original features, the overall FPR, accuracy and DR are presented in Table 3. Also, in Fig. 4, the Receiver Operating Characteristics (ROC) curves show the relationship between the DR and FPR with different K values. It is noted that the gradual increase in the K value from 2 to 10 with even numbers improves the overall DR and accuracy and decreases the overall FPR. When the K value increases gradually from 2 to 10, in the NSL-KDD dataset, the overall DR and accuracy increase from 92.1 to 98.1% and 92.3 to 98.8%, respectively, while the overall FPR reduces from 3.0 to 0.4%. Similarly, in the UNSW-NB15 dataset, the overall DR and accuracy increase from 75.1 to 91.0% and 77.4 and 91.8%, respectively, but the overall FPR decreases from 8.3 to 5.8%.

6.2.2 Performance on Principal Components

A summary of the performances of the GAA-ADS technique on the 15 principal components in terms of the overall FPR, accuracy and DR, is presented in Table 4 while, in Fig. 5, the ROC curves depict the DR and FPR with different K values. Corresponding to its performance on the original features, the increasing K value from 2 to 10 improves the overall DR and accuracy but reduces the overall FPR. However, the overall performance of the principal components is better than that on the original features by 1-2% because the principal components are designed on the basis of the highest variances which can vary between the areas of normal and attack records. In the NSL-KDD dataset, while the K value increases from 2 to 10, the overall DR and accuracy improve from 94.2 to 99.6% and 95.0 to 99.7%, respectively while, in contrast, the overall FPR decreases from 1.1 to 0.2%. Likewise, in the UNSW-NB15 dataset, the overall DR and accuracy increase from 75.4 to 91.3% and 76.6 to 92.8%, respectively, but the overall FPR decreases from 8.2 to 5.8%.

6.2.3 Comparisons of Performances

In the left-hand figure in Fig. 6, the ROC curves of the previous five evaluations using the two datasets show that the performances of the GAA-ADS technique on the principal

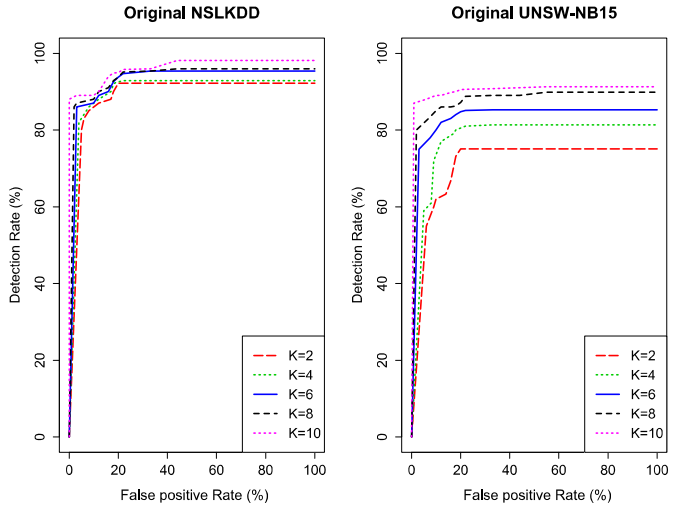

Fig. 4. ROC curves of original features using two datasets with different K values.

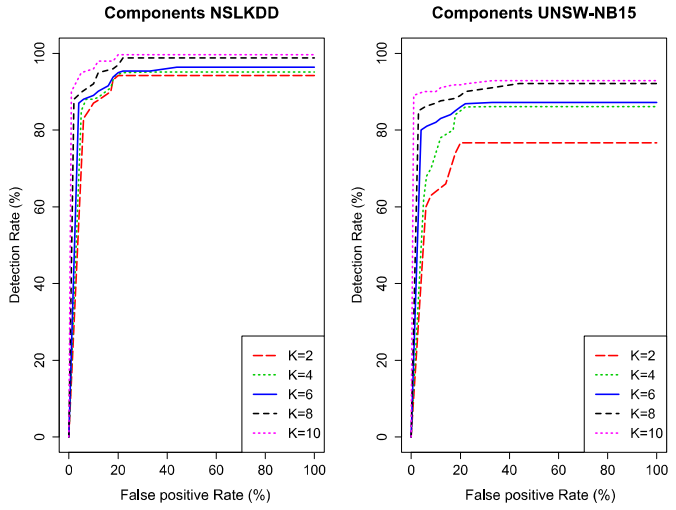

Fig. 5. ROC curves of principal components using two datasets with different k valuess.

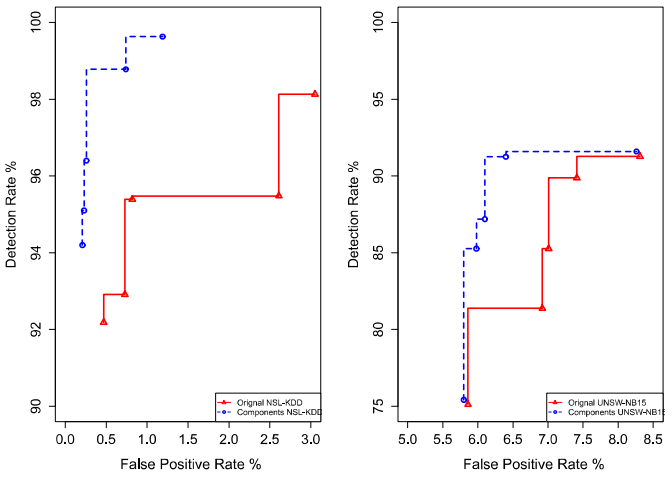

Fig. 6. ROC curves for two datasets.

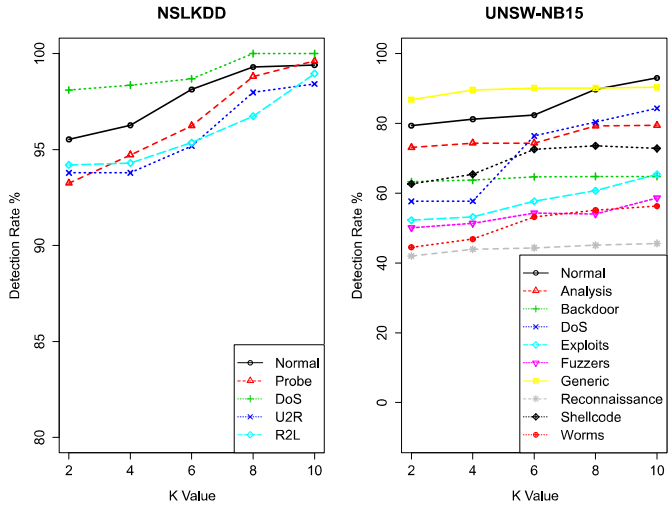

Fig. 7. Comparison of DR (%) on both datasets with K values.

components of the NSL-KDD dataset, which increase progressively from DRs of 94.2 to 99.6%, are better than those on the original features, which increase gradually from DRs of 92.1 to 98.1%, while the area of the FPR under the ROC curve of the principal components (i.e., [0.2 - 1.1 percent]) is nearly 50% of that of the original features (i.e., [0.4 - 3%]). The right-hand figure depicts the ROC curves of the original features and principal components using the UNSW-NB15 dataset which indicate that the difference between them is low since changes in their DR and FPR are approximately 0.3 and 0.7%, respectively.

Tables 5 and 6 show comparisons of the DRs of the record types for the K values on the components of the NSL-KDD and UNSW-N15 datasets, respectively, which indicate that, when the K value increases, the DR gradually improves, as shown by the comparisons in Fig. 7. The results in Table 5 demonstrate that the GAA-ADS technique can detect the majority of record types in the NSL-KDD dataset with a DR varying between 93.2 and 100% while the DRs of normal records increase from 95.5 to 99.2% with the lowest FN rate when the K value changes from 2 to 10. Similarly, the DRs of the attack types (i.e., Probe, DoS, U2R and R2L) increase gradually from, on average, 93.2 to 98.5%.

Table 6 indicates that the GAA-ADS technique detects record types in the UNSW-NB15 dataset with a DR varying from 42.0 to 93.0% while the DRs of normal records increase from 79.3 to 93.0% when the K value increases from 2 to 10. However, the DRs of the attack types do not always increase steadily; for instance, the Analysis, Backdoor, Fuzzers, Reconnaissance and Shellcode attacks do not achieve the best DRs with the highest K values while their differences from previous ones are low (approximately 1-2%) whereas the DRs of the other types of attack, DoS, Exploits, Generic and Worms, increase with increases in the K value. It can be observed that, as the variances of the components for these record types are close, the areas fall into each other in terms of decision-making.

Although this technique achieves better performances for the NSL-KDD dataset, those for some malicious records in the UNSW-NB15 one are not high because of the lower variances between them and normal records. As the UNSW-NB15 dataset was simulated using a sophisticated network architecture to include a contemporary broad range of malicious activities, our technique would be capable of protecting current networks while it is difficult to find all these activities in real networks by synchronously monitoring their different nodes. In [62], it was stated that deploying a collaborative ADS framework, in which each ADS cooperates with every other one to efficiently protect network nodes, could also help to improve the performance of the GAA-ADS technique in real networks.

The evaluation of the performances of the GAA-ADS technique is conducted using mainly two measures of the variances of features/principal components and K values. First, if the former are very high, the DR will be higher and the FPR lower. Based on the results, the principal components have the larger possible variances and always provide better detection accuracy than their original features because the variability of normal and anomaly observations

TABLE 5
Comparison of DR (%) on Components of NSL-KDD Dataset

Record type	K values				
	2	4	6	8	10
Normal	95.5%	96.2%	98.1%	99.3%	99.4%
Probe	93.2%	94.7%	96.2%	98.8%	99.6%
DoS	98.1%	98.3%	98.6%	100%	100%
U2R	93.7%	93.7%	95.1%	97.9%	98.4%
R2L	94.2%	94.3%	95.3%	96.7%	98.9%

TABLE 6
Comparison of DR (%) on Components of UNSW-NB15 Dataset

Record type	K values				
	2	4	6	8	10
Normal	79.3%	81.2%	82.3%	89.7%	93.0%
Analysis	73.1%	74.3%	74.3%	79.2%	79.4%
Backdoor	63.2%	63.7%	64.6%	64.8%	64.8%
DoS	57.6%	75.7%	76.4%	80.3%	84.3%
Exploits	52.2%	53.2%	57.6%	60.7%	65.4%
Fuzzers	50.1%	51.4%	54.3%	54.0%	58.6%
Generic	86.8%	89.6%	90.1%	90.1%	90.3%
Reconnaissance	42.0%	43.9%	44.3%	45.1%	45.6%
Shellcode	62.6%	65.4%	72.6%	73.6%	73.8%
Worms	44.5%	46.8%	53.1%	55.1%	56.2%

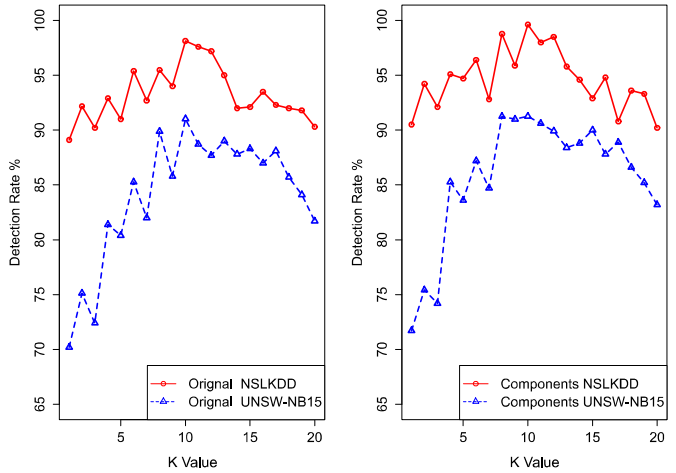

Fig. 8. Comparison of DRs (%) with 20 K values.

make their estimated areas different based on precise estimations of the TAEs. Since a TAE computes accurate areas of the features, we use it to estimate the total area of each record using the estimated BMM and distances between records because it can find small differences between the areas of the record types.

Second, gradual increases in the K value effectively improve performance. We tested the K values of all the possible values calculated using Equation (17) on both datasets. We observed that, for each two consecutive ranges (i.e., even and odd), when K equals even numbers from 2 to 10, the DRs of the even intervals are relatively better than those of the odd intervals, as shown in Fig. 8, in which the DRs and 20 K values are compared. Since the K even intervals from 2 to 10 include different variations of normal areas in small intervals, they considerably reduce the overlapping between normal and attack areas. However, by applying the other K values greater than 10, the DRs do not improve. Therefore, we should use only these K values from 2 to 10 to decrease the processing time and achieve higher DRs and lower FPRs.

6.2.4 Comparative Study and Discussion

We compare the performances of the GAA-ADS technique with those of six state-of-the-art anomaly detection techniques, namely, the Multivariate Correlation Analysis (MCA) [8], euclidean Distance Map (EDM) [63], Computer Vision Techniques (CVT) [62], Triangle Area Nearest Neighbours (TANN) [64], Artifical Immune System (AIS) [65] and Filter-based Support Vector Machine (FSVM) [32]. In Table 7, the results clearly reveal the superiority of our technique in terms of the DRs and FPRs on the NSLKDD dataset. The first four techniques were designed to detect only DoS attacks for which they achieve better detection but not for Probe, U2R and U2L attacks. Since these techniques depend on estimating the distances and correlations between normal and abnormal instances, as several attacks, especially stealth and spy attacks [4], significantly mimic normal observations, they interfere with the normal profile and decrease detection accuracy.

The AIS and FSVM techniques were designed to learn from normal and attack observations in the training phase based on the concept of rule-based learning. They always require a large number of observations to correctly learn

TABLE 7
Comparison of Performances of Six Techniques on NSL-KDD Dataset

Technique	DR	FPR
EDM[8]	94.2%	7.2%
MCA[63]	96.2%	4.9%
TANN[64]	91.1%	9.4%
CVT [62]	95.3%	5.6%
AIS [65]	90.4%	9.7%
FSVM [32]	92.2%	8.7%
GAA-ADS - original features	98.1%	0.4%
GAA - components	99.6%	0.2%

different patterns which it is very difficult to achieve in real networks. Their evalutions showed that their DRs are better for DoS and Probe attacks, of which there are sufficient instances but worse for rare events such as U2R and U2L attacks.

Ultimately, the GAA-ADS technique can achieve better performances than the other techniques for attack types with different K values, as demonstrated in Tables 2, 4, and 7, because the area of each record is accurately estimated and the K ranges include small differences that can efficiently discriminate between legitimate and malicious observations. This depends on estimating the BMM of the features and the distances between the records to simultaneously reflect the potential differences between attack and normal network traffic from data records and features.

6.2.5 Explanations of Complexity and Time Cost

We analyse the computational complexity and time cost of data processing for our proposed GAA-ADS technique. As described in Section 3, this method involves four steps: 1) computing the complexity of the PCA based on the Eigen decomposition of a covariance matrix which is $O(ND\times min(N,D))$ [59]; 2) computing the BMM parameters ($Beta(\pi,\upsilon,\omega)$); 3) computing the distance between the mean of normal records and each record ($distance_n$); and 4) determining the TAE for each record ($area(V)$). We combine the four big O notations for these steps to measure the total complexity of the proposed technique which processes N network records, each with D features.

On the one hand, in the training phase, the $Beta(\pi,\upsilon,\omega)$ parameters take $O(ND^3)$, with each computing the D features over N records. The $distance_n$ generates $O(ND)$ because all the records and features are processed only once, with the $area(V)$ taking $O(N)$ due to computing of all the features. On the other hand, the testing phase and decision-making generate $O(ND)$ due to applying the method of 'record by record' detection. As a result, for all records, the overall computational complexity of the technique is $O((ND\times\ min(N,D))+ND^3+ND+1)$. As the ND^3 term becomes larger than the other terms, the final overall computational complexity of the proposed technique is $O(ND^3)$. However as, like in [8], the D features are identically and independently distributed (iid) in each record and processed at the same time, the overall computational complexity is $O(1)$.

Comparing the four state-of-the-art approaches, the MCA technique [8] generates $O(ND^2)$ and $O(ND^4)$ in the training and testing phases, respectively and as, similar to our technique, its features are also iid, its overall

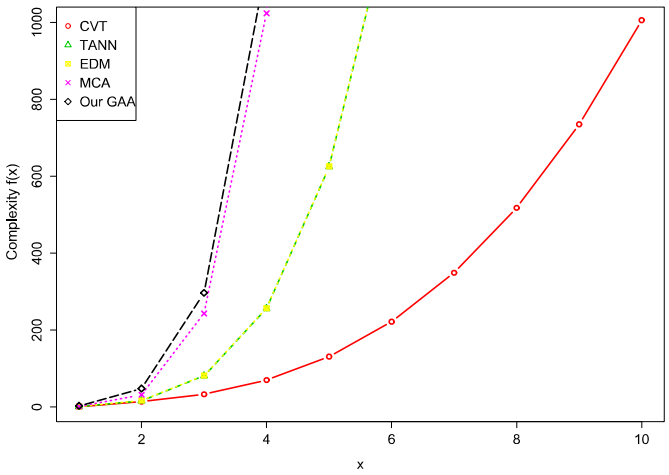

Fig. 9. Complexities of five techniques.

computational complexity is $O(1)$ while that of the EDM approach [63] is similar. These two techniques generate an overall computational complexity similar to that of the GAA technique which decreases to $O(D)$ because of computing the correlations between all possible combinations of the features. Another technique is the TANN [64] which is extremely complex as, in the training and testing phases, the computational complexities are $O(NDL^2)$ and $O(N^2DL^2)$, respectively, with L the number of clusters used to construct the triangular areas. Finally, the overall complexity of the CVT is $O((2ND^2)$ and $O(ND^4))$ in the training and testing sets, respectively [62], which is greater than that in our technique.

Fig. 9 shows the relationship among the five techniques using the mathematical equation of the big O notation $(f(x) = O(g(x))$, where $g(x)$ refers to the computed overall computational complexity and x to real numbers. As the number of records increases, the GAA-ADS technique executes faster than the other three approaches.

Furthermore, a time cost analysis demonstrates the influence of our GAA-ADS technique in terms of the speed of processing as it can execute approximately 23,696 records per second while the MCA, CVT and EDM approaches can process approximately 23,092, 19,267 and 12,044 records per second, respectively. In general, our algorithm can run 1.03, 2.17 and 10.27% faster than the MCA, CVT and EDM techniques, respectively.

6.2.6 Pros and Cons of GAA-ADS Technique

There are some advantages and limitations of the GAA-ADS technique. On the one hand, it is designed based on identifying the accurate areas of normal observations computed using the TAE and treating any deviation from them as an anomaly. The statistical methodology for building this technique verifies its capability to effectively detect existing and unknown attacks in large-scale networks as its profile involves only parameters which can automatically be updated by adjusting the K value. It also does not require any prior information about attack observations which shows its efficacy for application in online systems without any effort required in the training phase.

On the other hand, choosing the appropriate K value requires careful analysis of the data. In current networks, some types of attacks, such as stealth and spy ones, attempt to mimic normal activities [4]. First, we developed the GAA-ADS technique to find small variations between normal and anomaly areas but as, for current sophisticated attacks, their areas sometimes fall into those of normal ones, it is necessary to select the K value that efficiently differentiates between these areas. Second, as this technique can detect attacks without defining their types as it does not require any attack information in the training phase, attacks should be labelled by estimating their areas. Third, it can deal with only numeric features, an issue we overcame by using a feature conversion step in the pre-processing module. Finally because, if the variances between the selected features are not high, the performance of this technique will decrease, we applied the PCA to select the most highly varied features by testing these features and their principal components.

7 Conclusion and Directions for Future Work

This paper discussed a new GAA technique based on a BMM, distances between records and TAE to design an adaptive and lightweight anomaly detection system, with the estimated parameters and PDF of the BMM used to compute the TAE for each network traffic record. The TAE provided more accurate estimations using the BMM and distances between observations that facilitated discriminating between existing and new attacks in legitimate network traffic. This technique is used as a decision engine module at a proposed framework to establish a scalable, adaptive, and lightweight ADS. The evaluation of our technique' s performance and effectiveness was conducted using the NSL-KDD and UNSW-NB15 datasets involved in the data capturing and storing module.

The effect of selecting the highest variance data using the PCA for both the selected original features and their principal components was discussed in the data pre-processing module. The experimental results showed that the performance of the GAA-ADS technique was better on the principal components than on the original features. Also, the results from a comparative study of our technique and six state-of-the-art mechanisms in terms of detection accuracy, computational complexity and time cost revealed that the former outperformed the others in the overall DR, its computational complexity was equal to or better than them. In future, we will investigate new methods that could help the GAA to reduce the closed areas of attack types for eliminating the false alarm rate.

References

[1] R. Heady, G. F. Luger, A. Maccabe, and M. Servilla, *The Architecture of a Network Level Intrusion Detection System*. Albuquerque, NM, USA: Dept. Comput. Sci., College Eng., Univ. New Mexico, 1990.

[2] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Military Commun. Inf. Syst. Conf.*, 2015, pp. 1–6.

[3] S. Pontarelli, G. Bianchi, and S. Teofili, "Traffic-aware design of a high-speed FPGA network intrusion detection system," *IEEE Trans. Comput.*, vol. 62, no. 11, pp. 2322–2334, Nov. 2013.

[4] T. Giannetsos and T. Dimitriou, "Spy-sense: Spyware tool for executing stealthy exploits against sensor networks," in *Proc. 2nd ACM Workshop Hot Topics Wireless Netw. Secur. Privacy*, 2013, pp. 7–12.

[5] N. Moustafa and J. Slay, "The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Inf. Secur. J.: A Global Perspective*, vol. 25, no. 1–3, 2016. [Online]. Available: http://dx.doi.org/10.1080/19393555.2015.1125974

[6] W. Lee, S. J. Stolfo, and K. W. Mok, "A data mining framework for building intrusion detection models," in *Proc. IEEE Symp. Secur. Privacy*, 1999, pp. 120–132.

[7] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Comput. Secur.*, vol. 28, no. 1, pp. 18–28, 2009.

[8] Z. Tan, A. Jamdagni, X. He, P. Nanda, and R. P. Liu, "A system for denial-of-service attack detection based on multivariate correlation analysis," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 2, pp. 447–456, Feb. 2014.

[9] R. A. Redner and H. F. Walker, "Mixture densities, maximum likelihood and the EM algorithm," *SIAM Rev.*, vol. 26, no. 2, pp. 195–239, 1984.

[10] G. Creech and J. Hu, "A semantic approach to host-based intrusion detection systems using contiguousand discontiguous system call patterns," *IEEE Trans. Comput.*, vol. 63, no. 4, pp. 807–819, Apr. 2014.

[11] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: Methods, systems and tools," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 303–336, Jan.–Mar. 2014.

[12] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surveys*, vol. 41, no. 3, 2009, Art. no. 15.

[13] K. Lee, J. Kim, K. H. Kwon, Y. Han, and S. Kim, "DDoS attack detection method using cluster analysis," *Expert Syst. Appl.*, vol. 34, no. 3, pp. 1659–1665, 2008.

[14] J. Kim, P. J. Bentley, U. Aickelin, J. Greensmith, G. Tedesco, and J. Twycross, "Immune system approaches to intrusion detection–a review," *Natural Comput.*, vol. 6, no. 4, pp. 413–466, 2007.

[15] K. Ilgun, R. A. Kemmerer, and P. A. Porras, "State transition analysis: A rule-based intrusion detection approach," *IEEE Trans. Softw. Eng.*, vol. 21, no. 3, pp. 181–199, Mar. 1995.

[16] W. Hu, J. Gao, Y. Wang, O. Wu, and S. Maybank, "Online adaboost-based parameterized methods for dynamic distributed network intrusion detection," *IEEE Trans. Cybern.*, vol. 44, no. 1, pp. 66–82, Jan. 2014.

[17] A. Jamdagni, Z. Tan, X. He, P. Nanda, and R. P. Liu, "RePIDS: A multi tier real-time payload-based intrusion detection system," *Comput. Netw.*, vol. 57, no. 3, pp. 811–824, 2013.

[18] R. Mitchell and R. Chen, "Effect of intrusion detection and response on reliability of cyber physical systems," *IEEE Trans. Rel.*, vol. 62, no. 1, pp. 199–210, Mar. 2013.

[19] K. Singh, S. C. Guntuku, A. Thakur, and C. Hota, "Big data analytics framework for peer-to-peer Botnet detection using random forests," *Inf. Sci.*, vol. 278, pp. 488–497, 2014.

[20] M. D. Escobar and M. West, "Bayesian density estimation and inference using mixtures," *J. Amer. Statist. Assoc.*, vol. 90, no. 430, pp. 577–588, 1995.

[21] D. Wijayasekara, O. Linda, M. Manic, and C. Rieger, "Mining building energy management system data using fuzzy anomaly detection and linguistic descriptions," *IEEE Trans. Ind. Informat.*, vol. 10, no. 3, pp. 1829–1840, Aug. 2014.

[22] Z. Ma and A. Leijon, "Bayesian estimation of beta mixture models with variational inference," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 11, pp. 2160–2173, Nov. 2011.

[23] W. Fan, N. Bouguila, and D. Ziou, "Unsupervised anomaly intrusion detection via localized Bayesian feature selection," in *Proc. IEEE 11th Int. Conf. Data Mining*, 2011, pp. 1032–1037.

[24] B. Wagle, "Multivariate beta distribution and a test for multivariate normality," *J. Roy. Statist. Soc.. Series B (Methodological)*, vol. 30, pp. 511–516, 1968.

[25] A. K. Gupta and S. Nadarajah, *Handbook of Beta Distribution and Its Applications*. Boca Raton, FL, USA: CRC Press, 2004.

[26] N. Liao, S. Tian, and T. Wang, "Network forensics based on fuzzy logic and expert system," *Comput. Commun.*, vol. 32, no. 17, pp. 1881–1892, 2009.

[27] X. Yuan, Y. Meng, and X. Wei, "A method of location the vehicle windshield region for vehicle occupant detection system," in *Proc. IEEE 11th Int. Conf. Signal Process.*, 2012, vol. 1, pp. 712–715.

[28] S. Nedevschi, et al., "3D lane detection system based on stereovision," in *Proc. 7th Int. IEEE Conf. Intell. Transp. Syst.*, 2004, pp. 161–166.

[29] The NSLKDD dataset, May 2016. [Online]. Available: https://web.archive.org/web/20150205070216/http://nsl.cs.unb.ca/NSL-KDD/

[30] M. Tavallaee, E. Bagheri, W. Lu, and A.-A. Ghorbani, "A detailed analysis of the KDD Cup 99 data set," in *Proc. 2nd IEEE Symp. Comput. Intell. Secur. Defence Appl.*, 2009, pp. 1–6.

[31] *The UNSW-NB15 dataset*, May 2016. [Online]. Available: https://www.unsw.adfa.edu.au/australian-centre-for-cyber-security/cybersecurity/ADFA-NB15-Datasets/

[32] M. Ambusaidi, X. He, P. Nanda, and Z. Tan, "Building an intrusion detection system using a filter-based feature selection algorithm," *IEEE Trans. Comput.*, vol. 65, no. 10, pp. 2986–2998, Oct. 2016.

[33] C. J. Fung, Q. Zhu, R. Boutaba, and T. Ba, "Bayesian decision aggregation in collaborative intrusion detection networks," in *Proc. IEEE Netw. Operations Manage. Symp.*, 2010, pp. 349–356.

[34] W. Fan, N. Bouguila, and H. Sallay, "Anomaly intrusion detection using incremental learning of an infinite mixture model with feature selection," in *Proc. Int. Conf. Rough Sets Knowl. Technol.*, 2013, pp. 364–373.

[35] M. Gyanchandani, J. Rana, and R. Yadav, "Taxonomy of anomaly based intrusion detection system: A review," *Int. J. Sci. Res. Publications*, vol. 2, no. 12, pp. 1–13, 2012.

[36] P. Jongsuebsuk, N. Wattanapongsakorn, and C. Charnsripinyo, "Real-time intrusion detection with fuzzy genetic algorithm," in *Proc. 10th Int. Conf. Elect. Eng./Electron. Comput. Telecommun. Inf. Technol.*, 2013, pp. 1–6.

[37] J. Camacho, G. Maciá-Fernández, J. Diaz-Verdejo, and P. Garcia-Teodoro, "Tackling the big data 4 versus for anomaly detection," in *Proc. EEE Conf. Comput. Commun. Workshops*, 2014, pp. 500–505.

[38] Q. Ding and E. D. Kolaczyk, "A compressed PCA subspace method for anomaly detection in high-dimensional data," *IEEE Trans. Inf. Theory*, vol. 59, no. 11, pp. 7419–7433, Nov. 2013.

[39] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdisciplinary Rev.: Comput. Statist.*, vol. 2, no. 4, pp. 433–459, 2010.

[40] X. Xu and X. Wang, "An adaptive network intrusion detection method based on PCA and support vector machines," in *Proc. Int. Conf. Adv. Data Mining Appl.*, 2005, pp. 696–703.

[41] G. R. Zargar and T. Baghaie, "Category-based intrusion detection using PCA," *J. Inf. Secur.*, vol. 3, no. 4, 2012, Art. no. 259.

[42] E. De la Hoz, E. De La Hoz, A. Ortiz, J. Ortega, and B. Prieto, "PCA filtering and probabilistic SOM for network intrusion detection," *Neurocomputing*, vol. 164, pp. 71–81, 2015.

[43] M. A. T. Figueiredo and A. K. Jain, "Unsupervised learning of finite mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 3, pp. 381–396, Mar. 2002.

[44] Z. Ma and A. Leijon, "Beta mixture models and the application to image classification," in *Proc. 16th IEEE Int. Conf. Image Process.*, 2009, pp. 2045–2048.

[45] C. M. Bishop, "Pattern recognition," *Mach. Learn.*, vol. 128, pp. 1–19, 2006.

[46] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, 2nd ed, Berlin, Germany: Springer, vol. 12, pp. 100–250, 2013.

[47] T. Asano, T. Asano, and H. Imai, "Partitioning a polygonal region into trapezoids," *J. ACM*, vol. 33, no. 2, pp. 290–312, 1986.

[48] C. C. Aggarwal, "Outlier analysis," in *Data Mining*. Berlin, Germany: Springer, 2015, pp. 237–263.

[49] I. Ben-Gal, "Outlier detection," in *Data Mining and Knowledge Discovery Handbook*. Berlin, Germany: Springer, 2005, pp. 131–146.

[50] S.-H. Cha, "Comprehensive survey on distance/similarity measures between probability density functions," *City*, vol. 1, no. 2, 2007, Art. no. 1.

[51] The Ixia products, Jun. 2016. [Online]. Available: https://www.ixiacom.com/products/perfectstorm

[52] The MySQL cluster CGE technology, Jun. 2016. [Online]. Available: https://www.mysql.com/products/cluster/

[53] *The Hadoop technologies*, Aug. 2016. [Online]. Available: http://hadoop.apache.org/

[54] P. Pudil and J. Novovičová, "Novel methods for feature subset selection with respect to problem knowledge," in *Feature Extraction, Construction and Selection*. New York, NY, USA: Springer, 1998, pp. 101–116.

[55] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *J. Educ. Psychology*, vol. 24, no. 6, 1933, Art. no. 417.

[56] G. R. Zargar and P. Kabiri, "Identification of effective network features for probing attack detection," in *Proc. 1st Int. Conf. Netw. Digit. Technol.*, 2009, pp. 392–397.
[57] S. Chebrolu, A. Abraham, and J. P. Thomas, "Feature deduction and ensemble design of intrusion detection systems," *Comput. Secur.*, vol. 24, no. 4, pp. 295–307, 2005.
[58] S. Y. Sait, A. Bhandari, S. Khare, C. James, and H. A. Murthy, "Multi-level anomaly detection: Relevance of big data analytics in networks," *Sadhana*, vol. 40, no. 6, pp. 1737–1767, 2015.
[59] T. Elgamal, M. Yabandeh, A. Aboulnaga, W. Mustafa, and M. Hefeeda, "sPCA: Scalable principal component analysis for big data on distributed platforms," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2015, pp. 79–91.
[60] J. McHugh, "Testing intrusion detection systems: A critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln laboratory," *ACM Trans. Inf. Syst. Secur.*, vol. 3, no. 4, pp. 262–294, 2000.
[61] N. Moustafa and J. Slay, "The significant features of the UNSW-NB15 and the KDD99 data sets for network intrusion detection systems," in *Proc. 4th Int. Workshop Building Anal. Datasets Gathering Experience Returns Secur.*, 2015, pp. 25–31.
[62] Z. Tan, A. Jamdagni, X. He, P. Nanda, R. P. Liu, and J. Hu, "Detection of denial-of-service attacks based on computer vision techniques," *IEEE Trans. Comput.*, vol. 64, no. 9, pp. 2519–2533, Sep. 2015.
[63] Z. Tan, A. Jamdagni, X. He, P. Nanda, and R. P. Liu, "Denial-of-service attack detection based on multivariate correlation analysis," in *Proc. Int. Conf. Neural Inf. Process.*, 2011, pp. 756–765.
[64] C.-F. Tsai and C.-Y. Lin, "A triangle area based nearest neighbors approach to intrusion detection," *Pattern Recognit.*, vol. 43, no. 1, pp. 222–229, 2010.
[65] P. Saurabh and B. Verma, "An efficient proactive artificial immune system based anomaly detection and prevention system," *Expert Syst. Appl.*, vol. 60, pp. 311–320, 2016.

Nour Moustafa received the bachelor's and master's degrees of computer science from the Faculty of Computer and Information, Helwan University, Egypt, in 2009 and 2014, respectively. He is working toward the PhD degree at UNSW's Australian Centre for Cyber Security. His areas of interests include cyber security, in particular, network security, exploit defence, host- and network-intrusion detection systems, statistics, data mining, and machine learning techniques. He is a student member of the IEEE.

Jill Slay is a professor of Cyber Security and director of the Australian Centre for Cyber Security, UNSW Canberra @ ADFA. This centre has developed critical mass in cross-disciplinary research and teaching in Cyber Security to serve the Australian Government and Defence Force and help strengthen the Digital Economy. She is a member of the IEEE.

Gideon Creech received the BE (Hons.), MEngSc, MBA, and PhD degrees, as well as numerous industry qualifications including OSCP, OSCE, CISSP, and 10 GIAC certifications. He is a lecturer at UNSW's Australian Centre for Cyber Security. His research areas are network security, exploit development, host- and network-based intrusion detection systems, and cyber resilience for embedded systems. He is a member of the IEEE.