IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 19 December 2023, accepted 3 February 2024, date of publication 9 February 2024, date of current version 20 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3364400

RESEARCH ARTICLE

An Efficient Support Vector Machine Algorithm Based Network Outlier Detection System

OMAR ALGHUSHAIRY[1], RAED ALSINI[2], ZAKHRIYA ALHASSAN[1], ABDULRAHMAN A. ALSHDADI[1], AMEEN BANJAR[1], AYMAN YAFOZ[2], AND XIAOGANG MA[3]

[1]Department of Information Systems and Technology, College of Computer Science and Engineering, University of Jeddah, Jeddah 23890, Saudi Arabia
[2]Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia
[3]Department of Computer Science, University of Idaho, Moscow, ID 83844, USA

Corresponding authors: Omar Alghushairy (oalghushairy@uj.edu.sa) and Raed Alsini (ralsinie@kau.edu.sa)

This work was supported by the University of Jeddah, Jeddah, Saudi Arabia, under Grant UJ-22-DR-64.

ABSTRACT With the increase of cyber-attacks and security threats in the recent decade, it is necessary to safeguard sensitive data and provide robust protection to information systems and computer networks. In this paper, an anomaly-based network outlier detection system (NODS) is proposed and optimized to check and classify the incoming network traffic stream's behaviours that affect the computer networks. The proposed NODS has high classification efficiency. Network connection events classified as outliers are reported to the network admin to drop and block its packets. The NSL-KDD and CICIDS2017 intrusion datasets were employed to build the proposed system and test its detection capabilities. Sequential scenarios were implemented to optimize the system's effectiveness. Network features were normalized by min-max and Z-Score approaches, while the relevant features were selected individually by the principal component analysis (PCA) and correlated features selection (CFS) techniques. Support vector machine (SVM) and Gaussian Naive Bayes (GNB) algorithms are used to build the detection model, while the Genetic algorithm (GA) was employed to tune their control parameters. The obtained evaluation results proved that the proposed SVM based NODS is characterized by low false alarms and detection time as well as high classification accuracy. Furthermore, a comparative analysis was conducted with other existing techniques, and the results obtained demonstrate the effectiveness of the proposed SVM-IDS

INDEX TERMS Outlier detection, NSL-KDD, CICIDS2017, features normalization, features selection, support vector machine, Gaussian Naive Bayes, genetic algorithm, RBF, tunning parameters.

I. INTRODUCTION

Internet technologies and communication networks are evolving daily. In parallel, the advancement of cyber-attacks and the appearance of novel security vulnerabilities are quickly rising too [1]. Attempts that breach computer networks' availability, security, and privacy are known as network intrusions, anomalous or outliers [2]. Outlier detection is mainly employed for recognizing anomalous activities in many fields like network attacks detection. It is denoted as the process of identifying data points which are varied from the majority of other data [3]. These abnormal data points represent unusual behaviours and are denoted as outliers. A network outlier detection system (NODS) provides the mechanism to inspect network activities for detecting any possible intrusive actions [4]. NODS can be installed in a host such as a computer to audit its activities, including system calls and log files for detecting inclusive events [5]. Also, NODS can be deployed in a network to monitor and analyze its traffic stream behaviours to identify anomalous network connections [5]. Furthermore, NODS can identify intrusion attempts using the signature, anomaly, or hybrid-based detection approaches [6] The signature approach looks for the intrusion occurrence based on gathered knowledge

The associate editor coordinating the review of this manuscript and approving it for publication was Yudong Zhang.

about previous well-known intrusion signatures; therefore, it cannot identify the novel attacks [6]. The anomaly approach looks for any deviation from regular behaviour activities of a system or a network; therefore, it can recognize novel attacks. The hybrid approach integrates anomaly and signature-based detection methods to deliver a robust detection capability embedded in a single approach [6]. Regarding approaches used for detecting outliers, they are categorized as density, distance and machine learning or soft-computing [7]. In this paper, SVM and GNB are implemented individually to develop the anomaly-based detection model of NODS which is built and evaluated on the labelled network traffic stream of the benchmark NSL-KDD and CICIDS2017 datasets [8], [9]. Efficient data preprocessing of the network traffic data like features engineering is crucial in mitigating the model overfitting and boosting its generalization. Consequently, the outlier detection model performance gets improved and converged faster. The remainder of this paper is structured as follows: Section II reviews related work, Section III discusses the proposed NODS, Section IV highlights the implementation and results of the experiment. Finally, Section V presents the research conclusion along with the future interests.

II. RELATED WORKS

Network outliers are observations that are distinctly different from other observations, making them appear to be generated by a different process [10]. Unlike noise, network outliers carry important information, which can inform proactive network threat management. For example, an unusually large number of requests coming from one computer could be an outlier generated by a different process, which could indicate a malicious attack or some other type of unusual activity [11]. Thus, network outliers can help detect malicious behavior or provide insight into abnormal traffic patterns.

By detecting unusual activity in the network, organizations can identify malicious activities and reduce the risk of security breaches. Network anomaly detection can also be used to improve network performance by identifying and addressing network congestion, latency issues, and slow response times [11], [12]. Li et al. [13] developed an optimized resource allocation and communication technique for the fault detection system. This method is vital considering the limited edge device computation capabilities, minimal communication resources, and varying monitoring accuracies. The proposed approach maximizes the system's processing performance, optimizes resource use, and meets all data transmission and analysis latency needs.

From an organization's perspective, verifying the integrity of the network ensures that legitimate traffic is not blocked or rerouted to unknown sources, leading to a more secure and reliable network [14]. Pour et al. noted that by detecting anomalous activity, organizations can also ensure compliance with regulatory requirements, and improve the overall security posture of their network [15]. Furthermore, network anomaly detection can be used to monitor suspicious activity and detect potential malicious actors who may be attempting to gain access to the organization's network or data [14]. Thus, proactive network monitoring helps organizations to detect and respond to threats quickly, ensuring confidentiality, integrity, and availability of computing resources as well as preventing technical and business losses.

Lu et al. [16] address the issue of detecting the magnetic tile's internal defects leverages acoustic sound to detect the defects. The non-stationary and non-Gaussian properties of acoustic sound limit the accuracy of using a single data modality for detecting internal defects. Another study presents a novel ensemble and efficacious anomaly detection approach that relies on a collaborative representation-based detector. Background data is predicted using randomly chosen focused image pixels [17]. Connected and Autonomous Vehicles (CAVs) are becoming increasingly common due to the current technological development rate. However, these cars' networks are highly susceptible to illegal eavesdropping. Therefore, we propose using Deep Reinforcement Learning (DRL) and Distributed Kalman Filtering (DKF) methods to mitigate jamming interference and increase communication robustness to eavesdropping. The overarching aim is to optimize security performance against smart jammers and eavesdroppers. Thus, we formulate a DKF algorithm that accurately tracks the attacker by sharing state estimates between nodes. Consequently, we conceptualize a design problem for managing transmission power and picking communication channels. These provisions are made while ascertaining that the authorized vehicle user's quality needs are not compromised. A hierarchical Deep Q-Network (DQN)-based architecture is selected since the jamming and eavesdropping model is dynamic and uncertain. The DQN architecture is employed for designing channel selection policies and anti-eavesdropping power control. The optimal power control model is rapidly performed first without prior data or insights on eavesdropping behaviors. The channel selection process, which is founded on the system secrecy rate analysis, then proceeds when necessary. We simulate the proposed system, finding that it increases the secrecy and attainable communication rates [18].

Connected and Autonomous Vehicles (CAVs) are becoming increasingly common due to the current technological development rate. However, these cars' networks are highly susceptible to illegal eavesdropping. Therefore, we propose using Deep Reinforcement Learning (DRL) and Distributed Kalman Filtering (DKF) methods to mitigate jamming interference and increase communication robustness to eavesdropping. The overarching aim is to optimize security performance against smart jammers and eavesdroppers. Thus, we formulate a DKF algorithm that accurately tracks the attacker by sharing state estimates between nodes. Consequently, we conceptualize a design problem for managing transmission power and picking communication channels. These provisions are made while ascertaining that the authorized vehicle user's quality needs are not compromised. A hierarchical Deep Q-Network (DQN)-based

architecture is selected since the jamming and eavesdropping model is dynamic and uncertain. The DQN architecture is employed for designing channel selection policies and anti-eavesdropping power control. The optimal power control model is rapidly performed first without prior data or insights on eavesdropping behaviors. The channel selection process, which is founded on the system secrecy rate analysis, then proceeds when necessary. We simulate the proposed system, finding that it increases the secrecy and attainable communication rates [19].

Several practical challenges constrain the conventional ''forecast-response'' paradigm. For instance, the method's applicability is poor when different situations need dissimilar reaction processes. This deficiency originates from the paradigm's macro-perspective description of crises that overlooks the micro-perspective evaluation of emergency response. Therefore, this research recommends employing the ''scenario-response'' paradigm, which leverages a microscopic approach to frame the implications of conforming measures on events. Zhengzhou, China, experienced unexpected torrential rains in 2021 that resulted in 398 fatalities and approximately 120.6 billion RMB of economic losses. Consequently, an empirical assessment of the disaster based on Bayesian networks was done to analyze the emergency response's evolution. The constructed scenario Bayesian network was built by amalgamating Dempster's combination rule, scenario evolution, and knowledge meta-theory with 362 appropriate historical representative events. The network could also identify the progression of the respective emergency events and combine different experts' analyses. An event-driven Bayesian network was also employed to evaluate the impact of individual actions on the response outcomes' odds. The interventions' counterfactual outcomes were also checked using causal inference to highlight the urgent and vital responses. The similarity between each source and target scenario exceeded 0.7, with the highest value at 0.78. Furthermore, the incident response's evolutionary precision was examined by contrasting scenario parallels. Thus, the proposed approach can offer a theoretical foundation for deploying a ''scenario-response'' paradigm [20].

The number of multi objective large-scale optimization problems (MOLSOPs) has increased in recent years. The MOLSOPs can be addressed using cooperative coevolution and variable grouping optimization. However, few researchers have attempted to decompose MOLSOP variables. Therefore, they present a multi objective graph-based differential grouping with shift (mogDG-shift) for decomposing the multiple MOLSOP variables. We begin by assessing variable attributes and then detect the variable interactions. Consequently, we categorize the variables according to their interactions and features [21].

Asif et al. [22] developed an Intrusion Detection System (IDS), where KDD 99 intrusion dataset was used as the network traffic source. The detection system developed was designed to identify anomalous activities and network outliers early. Apache storm framework was used to handle the network stream big data characteristics. Assessment results stated the feasibility of the detection system. Besides, the system performance can be improved by solving the class imbalance problem. In [23], Han et al. developed an IDS to identify varied network attack types. Evolutionary neural networks (ENNs) were used to construct the detection model on the network traffic of the DARPA IDEVAL dataset. Evaluation results showed the system's ability in detecting network intrusion with low false alarms and a high detection rate. In [24], Wang et al. developed an IDS to complement the firewall. It can identify network attacks that the firewall cannot detect. The IDS was built based on the K-means clustering-based density and the k-NN classifier on the KDD intrusion dataset. Results proved that the system is effective in detecting varied network attacks. In [25], Sanjay et al. presented an improving mechanism for the attack detection system based on streaming data mining approaches. NSL-KDD intrusion dataset was used to assess four classification techniques, and their evaluation results are compared. Results proved that the Naïve Bayes classifier achieved the best accuracy, and the Hoeffding tree achieved the least detection time. In [26], Zhang et al. developed an outlier detection technique for data streams. The detection model is trained and assessed on KDD dataset. The performance evaluation proved the system's effectiveness in detecting network outliers at a lower rate of false positives than other compared systems.

Kurniabudi et al. utilized the Information Gain to rank and group features based on minimum weight values, enabling the selection of relevant and significant features [27]. Subsequently, we employ five classifier algorithms, namely Random Forest (RF), Bayes Net (BN), Random Tree (RT), Naive Bayes (NB), and J48, to conduct experiments on the CICIDS2017 dataset. The experimental results demonstrate that the number of relevant and significant features determined by Information Gain significantly impacts detection accuracy and execution time. Specifically, the Random Forest algorithm achieves the highest accuracy of 99.86% when using 22 relevant selected features, whereas the J48 classifier algorithm attains an accuracy of 99.87% with 52 relevant selected features, albeit requiring a longer execution time.

Pankaj Jairu et al. focused on building anomaly-based IDS to detect variety of network attacks by using many supervised learning algorithms such as Logistic Regression, Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Naïve Bayes, Decision Tree, and Random Forest on multiple datasets, including the realistic evaluation dataset CICIDS-2017 [28]. Results demonstrated that Random Forest outperformed other supervised algorithms and achieved an impressive accuracy of 99.93% by using only 14 features selected via Pearson's correlation coefficient method.

Shruti et al. introduced a novel intrusion detection system that employs ensemble techniques of machine learning algorithms [29]. The objective is to enhance classification accuracy and reduce false positives, utilizing

features sourced from the CICIDS-2017 dataset. The proposed presents an intrusion detection system (IDS) implemented through machine learning algorithms, including decision trees, random forests, and SVM. Additionally, this proposed incorporates LIME which is considered as an explainable framework to understand the model's prediction. The ensemble of ML models showed an improved accuracy of 96.25 for the IDS prediction, and the LIME explanation graphs showcased the prediction performance of the decision tree, random forest, and SVM algorithms. This integration aims to enhance comprehensibility and insight into the previously opaque black-box methodology for reliable intrusion detection.

Omar et. al have implemented five distinct deep learning models for the identification and categorization of suspicious activities within network flows in IOT environment [30]. These models are initially trained on a cloud server and subsequently deployed to a gateway node, where the pivotal network traffic classification is executed. The entire process of model training and assessment is conducted utilizing the CICIDS2017 dataset. The evaluation of the five models' accuracy revealed that the proposed model, named EIDM, exhibited exceptional performance, surpassing the other four models with a remarkable accuracy rate of 99.48%. This superior performance was achieved while also taking into consideration the time resources expended. Furthermore, the EIDM model proved its efficacy by successfully categorizing the full spectrum of 15 traffic behaviors, which encompassed 14 diverse attack types within the CICIDS2017 dataset, achieving a commendable accuracy level of 95%.

III. NETWORK OUTLIER DETECTION SYSTEM (NODS)

There are two main categories of NODS; supervised and unsupervised. If a system utilizes both supervised and unsupervised features, it is classified as semi-supervised [31] Supervised NODSs use labeled data to train a model that can then be used to detect outliers in new, unlabeled data sets. These systems are based on supervised learning techniques, such as decision trees, neural networks, and support vector machines (SVMs) [32]. These techniques are used to identify patterns in the data that indicate the presence of outliers. In decision tree-based NODSs, the data is split into multiple nodes based on the value of a certain feature [31], [33]. The nodes are then classified as outliers or inliers. Then, the system uses the decision tree to evaluate the data points and identify outliers.

Unsupervised NODSs use only unlabeled data to identify outliers. In this case, the dataset is first divided into two or more clusters, where each cluster represents a set of data points that share similar characteristics [32], [34]. The clusters are then evaluated to determine whether any data points are significantly different from the rest of the data. The evaluation is done using a variety of methods, such as density-based clustering, clustering based on distance, and cluster-based outlier detection algorithms. Once clusters are created, the next step is to identify anomalies in the data, which is achieved by calculating a score for each data point [22]. Consequentially, the score is calculated based on a variety of factors, such as distance from the cluster's central point, variance from the cluster's mean, and correlation with other data points in the cluster.

Smiti noted that if a data point has a significantly higher score than the rest of the data, it is considered an outlier [20]. Once outliers are identified, they can be further analyzed to determine what type of malicious activity is taking place [22]. As a result, the analysis can be done manually, or by using automated tools such as machine learning algorithms. NODS is deployed and attached to the entry point device of a computer network, as shown in Figure 1. Its goal is to capture and analyze the incoming network flow of this network.

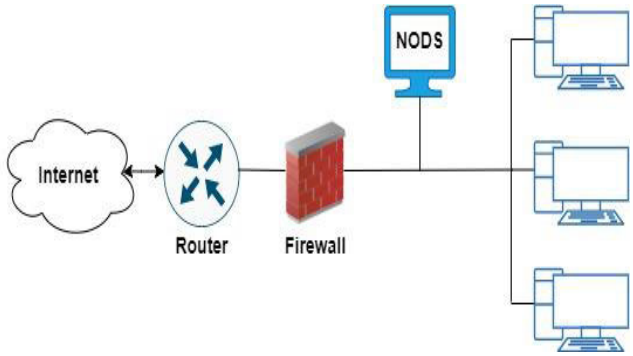

FIGURE 1. Network outlier detection system (NODS).

NODS starts with capturing the network traffic stream data by a packet sniffer. Then the related network packets are gathered to form numbers of network connections and generate them into a dataset file to be analyzed [35], [36], [37]. Each connection is described as a vector of many network features. Therefore, any network connection behaviour can be analyzed and classified as either normal or an outlier. Once NODS detects any abnormal network flow, an alarm is raised to the network admin to take suitable countermeasures regarding this outlier traffic, like dropping this anomalous traffic by blocking its IPs. However, processing these data directly represents long time analysis processes and leads to imprecise detection results. Therefore, it should be preprocessed well by many data mining techniques before being analyzed to ease the classification process and achieve efficient classification results.

A. NETWORK TRAFFIC DATA PRE-PROCESSING

1) NETWORK FEATURES ENCODING

The network features values are heterogeneous in their types where they can be founded either in nominal forms like protocol type, e.g. TCP or UDP, or in numeric form like a port number. Many outlier detection models cannot work with nominal data. It should be encoded into numeric form, and each connection's class/target feature is encoded to 0 for normal and 1 for the outlier/anomalous behaviour.

2) NETWORK FEATURES NORMALIZATION

Naturally, the values range of network features is varied, leading the outlier detection model for biasing toward the high scale features and ignoring others with a lesser scale. This results in an inaccurate detection process, which could lead to the model underfitting problem. Therefore, this problem is avoided by rescaling the values of the feature ranges on a uniform scale. Two normalization methods are used, the min-max and the Z-score.

a) Min-Max method scales each feature values between specific range of values [a,b] like [0 1] or [−1,+1] by the following formula

$$N(x) = a + \frac{(x - \min(x))(b - a)}{\max(x) - \min(x)} \quad (1)$$

where x is the original feature value, and N(x) denotes its normalized value.

b) Z-score method scales each feature according to its mean and standard deviation as the following formula

$$N(x) = \frac{(x) - mean(x)}{std(x)} \quad (2)$$

3) NETWORK FEATURES SELECTION

The network connection is described as a vector of network features representing the connection behaviour. The information contribution of these features concerning the connection behaviour label is varied [38]. Many features hold less information about the connection behaviour denoted by irrelevant features, while others contain redundant information denoted by redundant features. Building the detection model on either irrelevant or redundant features causes the overfitting problem rather than increasing the model complexity [39]. Discarding those features during the model building process improves model classification capabilities [39]. Two features selection techniques, PCA [40] and CFS [41] are adopted to select the dominant features from the whole network features set for building the detection model on its basis. PCA selects a subset of network features that has the higher eigenvalues. In contrast, CFS selects features with a high correlation with the class/label of the network connection behaviour and low or no correlation between each other.

B. NETWORK OUTLIER DETECTION

1) SVM MODELS FOR NODS

Support Vector Machines (SVMs) are a type of supervised learning algorithm that has been successfully applied to a variety of classification and regression problems. The SVM algorithm is based on the idea of finding a hyperplane that best separates the data points into two distinct classes. The SVM algorithm seeks to maximize the margin between the two classes, thereby obtaining a ''maximum-margin hyperplane'' [42]. This hyperplane is determined through a process of optimization which minimizes the overall classification error. In SVM models, support vectors form the basis of the decision boundary which separates the two classes and has the maximum influence on the position of the hyperplane [42], [43].

TABLE 1. NODS implementation general algorithm.

Algorithm 1: **Network Outlier Detection System**
Input: *NSL-KDD OR CICIDS2017 Training and Testing dataset*
Output: *Network Connection behaviour Class*
(1) Pre-processing the input network connections data - Encoding the nominal features - Scaling all features value by Zscore and Min-Max [-1:1]
(2) Select the most informative network features subset by PCA and CFS.
(3) Train individually the SVM and GNB as detection models on the selected features subset of each PCA and CFS.
(4) Getting the generalized detection model.
(5) Use SVM or GNB detection model to classify the unseen network connections into begin and anomaly.

SVM models are applied in NODS because, in these systems, the goal is to identify ''outliers''—data points that are significantly different from the data points in the same class or cluster. Outliers can indicate malicious behavior, faulty or malfunctioning nodes, or other anomalies [44]. To detect these outliers, it is necessary to use an algorithm that can distinguish between normal and abnormal data points. SVMs models are well-suited for this task because they are capable of finding non-linear boundaries between data points.

SVM is considered a good candidate for building the anomaly-based outlier classification model. It begins with learning the network traffic's normal/usual/inlier behaviour obtained from the previous preprocessing stage. After, it builds a model which can recognize both normal and abnormal behaviours of unseen network traffic. Each network connection differs from the usual behaviour/pattern treated as an outlier connection.

2) GAUSSIAN NAIVE BAYES (GNB) MODEL FOR NODS

Considered a popular supervised probabilistic algorithm model and based on Bayes' theorem. It is commonly used for text classification and is widely used in various machine-learning tasks, including spam filtering, intrusion detection, and sentiment analysis [37].

The key assumption in GNB is that all features are conditionally independent given the class label. In other words, it assumes that the presence or absence of a particular feature does not affect the presence or absence of other features in the same class. This is a strong and often unrealistic assumption, but it allows the algorithm to be computationally efficient and work well with high-dimensional data [45].

GNB is an effective choice for identifying anomalous network activities and potential security threats. By considering the statistical distribution of features related to network traffic, such as packet sizes, response times, and connection

duration, the model can learn patterns of normal behavior. During the testing phase, it can efficiently classify incoming data as either normal or malicious based on the learned probability distributions [46].

3) TUNING SVM AND GNB CONTROL PARAMETERS BY USING GA

Radial Basis Function (RBF) SVMs are becoming increasingly popular for classification, regression, and clustering tasks such as network outlier detection. Wainer et al. noted that RBF technique is preferred due to its capability to map non-linear data, which allows them to capture complex patterns in the data [41].

SVM uses the RBF as a kernel function during the classification process. RBF has two parameters: the penalty (c) and kernel parameter (σ). The former controls the SVM's hyperplane flexibility, while the latter controls the correlation among support vectors of the same hyperplane. These parameters have an observable impact on the SVM classification effectiveness. Thus, it's necessary to properly tune these parameters values which considered an optimization problem.

For the GNB, the primary parameter that can be adjusted is the smoothing parameter which is used to prevent zero probabilities when a particular feature value is not observed in the training data for a given class. The smoothing parameter is a positive value added to all feature occurrences, which helps in handling unseen feature combinations and avoids division by zero in probability calculations [47].

Genetic Algorithms (GAs) have become an increasingly popular tool for optimizing complex systems, including NODS. GAs have been shown to outperform traditional optimization techniques in a variety of applications, from distributed systems to clustering algorithms. GAs also provide efficient and robust solutions for outlier detection, with applications in network intrusion detection, fraud detection, and traffic anomaly detection [48]. Notably, traditional methods of NODS rely on static rules and thresholds, which can be difficult to maintain and may not always be accurate.

GAs offer an alternative approach to NODS, providing a more dynamic and adaptive solution. The basic idea behind GAs is to use evolutionary algorithms to search for the best solutions to a given problem. In the case of network outlier detection, this means using GAs to optimize the parameters and thresholds used to detect outliers [49]. GAs are able to search through a large and complex search space to identify the best parameters for a given problem. In this research, GA employed to search for the best values of RBF parameters in this research, GA is employed to search for the best values of SVM's RBF and GNB's smoothing parameters in a given search space which consists of number of candidates each representing possible values for these parameters. Determining the appropriate candidate will boost SVM and GNB detection performance. Further theoretical and technical details on SVM, GNB and GA techniques are discussed in [45], [50], [51], and [52].

IV. IMPLEMENTATION AND EXPERIMENTAL RESULTS

A. NETWORK INTRUSION DATASET

1) NSL-KDD is a benchmark labelled network traffic dataset used globally by researchers who are interested in intrusion detection field area [53]. It consists of two files, the training set with 127973 network connection instances and the testing set with 22544. Each connection described by a vector of 42 features as mentioned in Table 1. For the feature value types, all are considered as numeric except feature numbers (2,3,4,42) are nominal, as shown in Table 1. The behaviour of each connection is classified as either normal or outlier.

It has 38 varied attack types, where the training set contains 22 types, and the testing set involves the other 16 [39]. Table 3 groups these attacks into four categories as following:

2. Probe: Intruder aims to obtain varied information concerning the victim host or network by scanning its opened and closed ports, rather than its IPs ranges to launch future attacks.
3. Denial of Service: By using zombies, intruders can flood the target system with huge numbers of network packets. As a sequence, the victim system resources e.g. network bandwidth, and processing power are exhausted and become unreachable for its legitimated users.
4. User to Root: Intruder aims to acquire the root/admin privileges of the victim machine by exploring and exploiting their vulnerabilities.

1) Remote to Local: Intruder who has no account on the host aims to get unauthorized access to it.

2) CICIDS2017 is a benchmark dataset widely used in the field of intrusion detection research [54]. It was created to evaluate the performance of IDS in accurately identifying network attacks and distinguishing them from legitimate network activities. Most of the available network traffic datasets suffer from the absence of traffic diversity, volumes, anonymized packet information payload, constraints on the attacks range, the lack of the feature set and metadata. Therefore, this dataset came to conquer these concerns. It comprises various types of network traffic, including benign/normal traffic and different categories of attacks including Brute Force attack, Web attack, DoS, Infiltration, Botnet, PortScan and DDoS. It consists of 2830540 connection instances where each is described by a vector of 79 features as mentioned in Table 4. All network traffic flow classes categorization of the CICIDS2017 dataset are listed in Table 5, where all detailed analysis of the CICIDS2017 dataset is existed at [55].

B. EXPERIMENTAL SETUP

A personal laptop is used to carry the proposed research experiments with 4 GB RAM, Intel core i7 CPU, and Window 10 OS. The setup of these experiments was as follow:

TABLE 2. NSL-KDD dataset network features list.

Feature Number	Network Feature Name
1	Duration
2	Protocol_type
3	Service
4	Flag
5	Src_bytes
6	Dst_bytes
7	Land
8	Wrong_fragment
9	Urgent
10	Hot
11	Num_failed_logins
12	Logged_in
13	Num_compromised
14	Root_shell
15	Su_attempted
16	Num_root
17	Num_file_creations
18	Num_shells
19	Num_access_files
20	Num_outband_cmds
21	İs_hot_login
22	Is_guest_login
23	Count
24	Srv_count
25	Serror_rate
26	Srv_serror_rate
27	Rerror_rate
28	Srv_rerror_rate
29	Same_srv_rate
30	Diff_srv_rate
31	Srv_dif_host_rate
32	Dst_host_count
33	Dst_host_srv_count
34	Dst_host_same_srv_rate
35	Dst_host_diff_srv_rate
36	Dst_host_same_src_port_rate
37	Dst_host_srv_dif_host_rate
38	Dst_host_serror_rate
39	Dst_host_srv_serror_rate
40	Dst_host_rerror_rate
41	Dst_host_srv_rerror_rate
42	Class label

- Min-max and Z-score scaler/normalizer techniques are implemented in Python to normalize and rescale the input feature values of network traffic data.
- The Java-based weka platform is used to implement the features selection process from network traffic data by two filter techniques PCA and CFS.

TABLE 3. All 38 attack types with four classes of NSL-KDD dataset.

Class	Attack name
Probe	Portsweep, Saint, Satan, Nmap, Ipsweep, Mscan
Denial of Service (Dos)	Land, Smurf, Apache2, Udpstorm, Neptune, Processtable, Pod, Worm, Teardrop
User to Root (UTR)	Rootkit, Loadmodule, Sqlattack, Ps, Xterm, Perl, Buffer_overflow
Remote to Local (RTL)	Warezmaster, Phf, Xsnoop, Ftp_write, Sendmail, Guess_Password, Named, Httptunnel, Imap, Xlock, Snmpguess, Warezclient, Snmpgetattack, Spy, Multihop

- The Python-based Scikit-learn machine learning library is employed for implementing and building the SVM and GNB detection models individually on the network traffic data of the NSL-KDD and CICIDS2017 datasets and adopts the superiority of them as the detection model for the proposed NODS.
- GA is implemented in Python to adjust and tune RBF control parameters by using SVM and the smoothing parameter of the GNB models. The model detection accuracy is used as the GA fitness function for evaluating each candidate/individual/chromosome fitness during the GA generation process.
- The number of GA iterations was 100, and the size of the GA population was 300 candidates. Each GA candidate consists of either two random values for SVM RBF [penalty parameter (c), kernel parameter (σ)] or one random value for the GNB's smoothing parameter.
- The range values for the SVM RBF [penalty, kernel] and GNB smoothing parameter are [.01:4000,.01:100], and [.01:100] respectively.
- For the NODS implementation, 125973 and 22543 instances from NSL-KDD are used for the training and testing steps, while 120023 and 30006 instances are used from the CICIDS2017, respectively.
- The overall performance of the SVM and GNB detection models is evaluated individually on the NSL-KDD and CICIDS2017 datasets by many evaluation metrics as discussed in the next subsection.

C. PERFORMANCE EVALUATION METRICS

Many metrics are calculated to evaluate the capabilities of the proposed NODS. These metrics are inferred from the following confusion matrix:

		Detected Class	
		Positive / Attack	**Negative / Normal**
Actual Class	**Positive / Attack**	True positive (TP)	False Negative (FN)
	Negative / Normal	False Positive (FP)	True Negative (TN)

TABLE 4. CICIDS2017 dataset network features list.

Feature No.	Network Feature Name
1	Destination Port
2	Flow Duration
3	Total Fwd Packets
4	Total Backward Packets
5	Total Length of Fwd Packet
6	Total Length of Bwd Packet
7	Fwd Packet Length Max
8	Fwd Packet Length Min
9	Fwd Packet Length Mean
10	Fwd Packet Length Std
11	Bwd Packet Length Max
12	Bwd Packet Length Min
13	Bwd Packet Length Mean
14	Bwd Packet Length Std
15	Flow Bytes/s
16	Flow Packets/s
17	Flow IAT Mean
18	Flow IAT Std
19	Flow IAT Max
20	Flow IAT Min
21	Fwd IAT Total
22	Fwd IAT Mean
23	Fwd IAT Std
24	Fwd IAT Max
25	Fwd IAT Min
26	Bwd IAT Total
27	Bwd IAT Mean
28	Bwd IAT Std
29	Bwd IAT Max
30	Bwd IAT Min
31	Fwd PSH Flags
32	Bwd PSH Flags
33	Fwd URG Flags
34	Bwd URG Flags
35	Fwd Header Length
36	Bwd Header Length
37	Fwd Packets/s
38	Bwd Packets/s
39	Min Packet Length
40	Max Packet Length
41	Packet Length Mean
42	Packet Length Std
43	Packet Length Variance
44	FIN Flag Count
45	SYN Flag Count
46	RST Flag Count
47	PSH Flag Count
48	ACK Flag Count
49	URG Flag Count
50	CWE Flag Count
51	ECE Flag Count
52	Down/Up Ratio
53	Average Packet Size
54	Avg Fwd Segment Size
55	Avg Bwd Segment Size
56	Fwd Header Length
57	Fwd Avg Bytes/Bulk
58	Fwd Avg Packets/Bulk
59	Fwd Avg Bulk Rate
60	Bwd Avg Bytes/Bulk
61	Bwd Avg Packets/Bulk
62	Bwd Avg Bulk Rate
63	Subflow Fwd Packets
64	Subflow Fwd Bytes
65	Subflow Bwd Packets
66	Subflow Bwd Bytes
67	Init_Win_bytes_fwd
68	Init_Win_bytes_bwd
69	Act_data_pkt_fwd
70	Min_seg_size_fwd
71	Active Mean
72	Active Std
73	Active Max
74	Active Min
75	Idle Mean
76	Idle Std
77	Idle Max
78	Idle Min
79	Label

All evaluation metrics are detailed as following [56]:

1. Detection Accuracy (DC): denotes the proportion of the properly detected network connections to whole detected connections.

$$DC = \frac{TN + TP}{FN + FP + TN + TP} \tag{3}$$

2. Detection Rate (DR): denotes the ratio of the properly predicted network connections as outliers to the whole real outlier connections.

$$DR = \frac{TP}{TP + FN} \tag{4}$$

3. False Negative Rate (FNR): denotes the ratio of the outlier network connections wrongly identified as normal to the whole real outlier connections.

$$FNR = \frac{FN}{FN + TP} \tag{5}$$

TABLE 5. Network traffic class composition of the CICIDS2017 dataset.

Traffic Status	Traffic Label
Normal	Benign
Bot	BOT
Brute Force	FTP-Patator SSH-Patator
DOS/DDOS	DOS DDOS DoS GoldenEye DoS Hulk DoS SlowHttpTest DoS slowloris Heartbleed
Infiltration	Infiltration
PortScan	PortScan
Web Attack	Web Attack-Brute Force Web Attack-Sql Injection Web Attack-XSS

4. False Positive Rate (FPR): denotes the ratio of the normal network connections wrongly identified as outliers to all real normal connections.

$$FPR = \frac{FP}{FP + TN} \tag{6}$$

5. Detection time (DT): represents the time taken to classify the behaviours of all unseen network connections existed in the testing file of the dataset.
6. Area Under the Curve (AUC): measures the NODS performance in identifying the normal and outlier classes.

D. EXPERIMENT SCENARIOS AND RESULTS DISCUSSION

Proposed research experiments are conducted by carrying out four scenarios for developing and optimizing the proposed NODS. The first scenario mimics building the detection system on the original network traffic data of the pre-mentioned dataset without performing any data preprocessing stages. The second scenario mimics performing only one data preprocessing stage by normalizing the network traffic data by min-max [-1:+1], and z-score scaler methods before building the detection system. The third scenario mimics applying two data preprocessing stages before building the detection system.

After normalizing the input network traffic data by the best scaler approach determined from the previous scenario, we apply the dimensionality reduction process on the input normalized data by selecting the most informative and significant features subset from the whole features set. Two filter feature selection techniques, the PCA and IG, are applied individually on the input normalized network data before the learning process to detect which selection technique affect positively the NODS detection performance. Finally, the fourth scenario mimics employing GA to tune the hyperparameters of the SVM's RBF control parameters [c, σ] and the smoothing parameter of the GNB during the building process of the used detection model on the pre-selected network features subset obtained from the previous scenario and analyze their impact on the final performance of the proposed NODS. For the GA setup, we noticed that using large individuals/candidates' numbers of the GA population resulted in providing better genetic variability and a faster adaptation as well. And based on many pre-empirical experimental tests and trials, we set the number of individuals in the GA population to 300, and the generations number to 100.

Concerning the first scenario, the SVM and GNB detection models performance built on both the NSL-KDD and CICIDS2017 datasets are ineffective totally according to their evaluation results shown in Table 6 and 7. Due to the low quality and non-preprocessing of the input network data, the detection model got a high underfitting. Therefore, both detection models' accuracy and detection rates in recognizing the network traffic were very low, and they required a long time for classifying the traffic behaviour. As a result, the network admin will be confused about the high false alarm rates because much intrusive network traffics are recognized as normal.

TABLE 6. The NODS performance evaluation of the first scenario on NSL-KDD dataset.

Performance metrics	SVM-NODS	GNB-NODS
DC (%)	**70.73**	45
DR (%)	**59.78**	3.6
FNR (%)	**40.21**	96.3
FPR (%)	**3.01**	0.32
DT (sec)	**367.9**	0.026
AUC	**0.74**	0.51

TABLE 7. The proposed NODS performance evaluation of the first scenario on CICIDS2017 dataset.

Performance metrics	SVM-NODS	GNB-NODS
DC (%)	65.5	33.64
DR (%)	12.47	23.02
FNR (%)	87.52	76.97
FPR (%)	21.65	2.83
DT (sec)	195.2	0.007
AUC	0.45	0.576

For the second scenario, applying the min-max [−1:1] and z-score normalization techniques as a data preprocessing task to rescale and normalize the input network features values before the detection model training process. It helps in preventing the biasing problem occurrence to the detection

model toward the network features with high scale values where this problem always affects negatively the model performance. As shown in Table 8,9, and figures 2,3, both SVM and GNB detection models performance after applying the min-max [−1:1], and z-score methods were better than the performance of the first scenario detection model. Results ensure the importance of applying the normalization task during the data preprocessing stage before the learning process starts. Regarding the impact of the two normalization approaches used for enhancing the SVM and GNB detection models performance, the impact of applying z-score outperformed the min-max [-1:1] scaler method on the models built on the network traffic data of the NSL-KDD dataset where the vice versa on the CICIDS2017 dataset. So, applying the normalization task helps in overcoming the model biasing and underfitting problems and therefore optimizing the NODS capabilities to be more effective and faster. In addition, the detection model misclassifying rates represented in either the false negative or positive alarms became much lower than the first scenario results.

TABLE 8. The proposed NODS performance evaluation of the second scenario on NSL-KDD dataset.

Performance metrics	NODS Evaluation Results on NSL-KDD			
	SVM	GNB	SVM	GNB
Features Selection technique	-1:+1		ZScore	
DC (%)	77.5	57.8	**98.3**	**88**
DR (%)	66.2	51.3	**97.9**	**89**
FNR (%)	33.7	48.6	**2**	**10.9**
FPR (%)	2.8	38.8	**1.2**	**12.9**
DT (sec)	32.8	0.02	**63.1**	**0.02**
AUC	0.79	0.55	**0.98**	**0.88**

Regarding the third scenario, applying the dimensionality reduction task on the best-normalized network traffic features from both NSL-KDD and CICIDS2017 data resulted from the previous scenario. Two common feature selection techniques, PCA and CFS, are applied individually on the normalized data before the SVM and GNB detection models learning process, to assess their impact on the overall detection capabilities of the used models.

The selected feature subsets from both the zscore-based NSL-KDD and min-max [-1:1] based CICIDS2017 are tabulated with their indices in Table 10,11. Both SVM and GNB detection models are built on these selected feature subsets and their evaluation performance is evaluated. Results in Table 12,13 stated that the PCA technique outperformed CFS in selecting the most relevant and informative features from both the used two datasets. Consequently, it led for achieving a significant contribution in decreasing the SVM and GNB detection models learning time, complexity, and

TABLE 9. The proposed NODS performance evaluation of the second scenario on CICIDS2017 dataset.

Performance metrics	NODS Evaluation Results on CICIDS2017			
	SVM	GNB	SVM	GNB
Features Selection technique	-1:+1		ZScore	
DC (%)	**68.13**	**40.07**	66.04	37.05
DR (%)	**18.3**	**24.72**	13.33	23.79
FNR (%)	**81.69**	**75.27**	86.66	76.2
FPR (%)	**19.82**	**1.66**	21.05	0.88
DT (sec)	**69.67**	**0.022**	0.025	0.025
AUC	**0.49**	**0.61**	0.76	0.6

mitigating the overfitting risk. Furthermore, accelerating the detection models time, and improving their effectiveness in analyzing the input network traffic behaviours compared with the second scenario results.

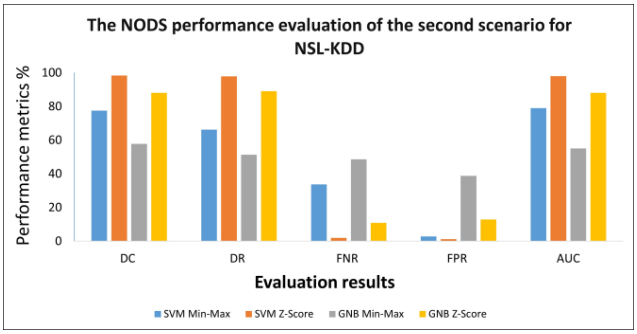

FIGURE 2. NODS performance on the second scenario for NSL-KDD using the Min-max, and Z-score.

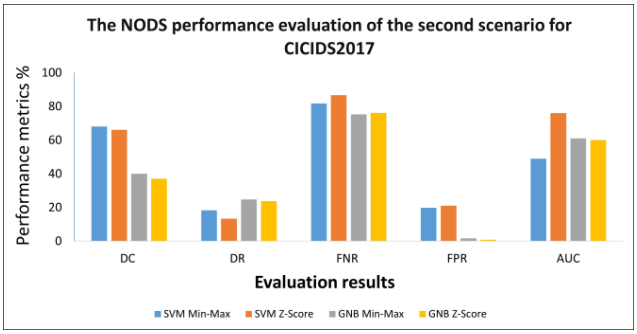

FIGURE 3. NODS performance on the second scenario for CICIDS2017 using the Min-max, and Z-score.

Regarding the fourth scenario, the GA is used to tune the RBF control parameters [c, σ] of the SVM and the smoothing parameter of the GNB during their learning process on

TABLE 10. Selected features subset by the CFS, and PCA techniques.

Selection Algorithm	Features subset length	Selected Features
CFS	9	2, 3, 6, 8, 13, 17, 21, 22, 27
PCA	22	1, 2, 3, 9, 11, 22, 24, 25, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40

TABLE 11. Selected CICIDS2017's features subset by the CFS, and PCA techniques.

Selection Algorithm	Features subset length	Selected Features
CFS	17	1,5,6,13,14,17,24,35,38,41,42,43, 53,56,64,67,68
PCA	20	1,2,3,5,6,14,17,19,24,25,35,36,4 0,41,42,43,53,55,56,67

TABLE 12. The proposed NODS performance evaluation of the third scenario on NSL-KDD dataset.

Performance metrics	NODS Evaluation Results on NSL-KDD			
	CFS-SVM	PCA-SVM	CSF-GNB	PCA-GNB
Features Length	9	**22**	9	**22**
DC (%)	92.3	**98.6**	85.13	**87.74**
DR (%)	90.53	**99.08**	82.39	**87.06**
FNR (%)	9.46	**0.9**	17.6	**12.9**
FPR (%)	5.4	**1.8**	11.2	**11.4**
DT (sec)	75.26	**78.17**	0.006	**0.012**
AUC	0.92	**0.98**	0.84	**0.87**

the previous PCA-based selected network features of the used datasets from the last scenario. Results in Table 14,15 stated that adjusting the two detection models hyperparameters resulted in boosting their generalization ability and convergence speed which led to an optimization in the overall performance of the SVM and GNB models.

Regarding the evaluation comparison between the four successive scenarios, it's noted that the fourth detection NODS models (PCA-GA-SVM and PCA-GA-GNB) considered the superlative among all previous NODS scenarios in detecting the normality and abnormality behaviours of the network traffic connections of the used datasets.

For a comparison with other related detection systems as shown in Table 16, evaluation results stated the superiority

TABLE 13. The proposed NODS performance evaluation of the third scenario on CICIDS2017 dataset.

Performance metrics	NODS Evaluation Results on CICIDS2017			
	CFS-SVM	PCA-SVM	CSF-GNB	PCA-GNB
Features Length	17	**20**	17	**20**
DC (%)	77.24	**81.03**	80.01	**87.03**
DR (%)	42.38	**52.46**	49.16	**79.67**
FNR (%)	57.61	**47.53**	50.8	**20.32**
FPR (%)	14.4	**12.7**	12.67	**12.04**
DT (sec)	124.12	**43.24**	0.008	**0.009**
AUC	0.63	**0.68**	0.67	**0.712**

TABLE 14. The proposed NODS performance evaluation of the fourth scenario on NSL-KDD dataset.

Performance Metrics	NODS Evaluation Results on NSL-KDD	
	PCA- GA -SVM	PCA- GA -GNB
Features Length	**22**	
Hyper Parameters	**[4, 0.16]**	**[0.09]**
DC (%)	**99.25**	**88.21**
DR (%)	**99.48**	**86.8**
FNR (%)	**0.51**	**13.17**
FPR (%)	**0.99**	**10.15**
DT (sec)	**30.07**	**0.013**
AUC	**0.99**	**0.88**

TABLE 15. The proposed NODS performance evaluation of the fourth scenario on CICIDS2017 dataset.

Performance Metrics	NODS Evaluation Results on CICIDS2017	
	PCA- GA -SVM	PCA- GA -GNB
Features Length	**20**	
Hyper Parameters	**[0.9, 0.01]**	**[1.5]**
DC (%)	**88.43**	**88.74**
DR (%)	**96.86**	**98.82**
FNR (%)	**3.13**	**1.17**
FPR (%)	**12.36**	**12.19**
DT (sec)	**57.5**	**0.007**
AUC	**0.710**	**0.714**

of our proposed system with lower false alarms and higher detection accuracy.

TABLE 16. The Proposed NODS evaluation performance comparison with other related work.

Metrics	Proposed	[58] 2019	[59] 2019	[60] 2020	[61] 2021	[62] 2023	[63] 2023
DC	**99.25**	93.9	91.7	98.81	99	98.8	98.96
DR	**99.48**	95.5	92.4	97.25	99.1	98.87	96.13
FNR	**0.5**	4.5	7.6	2.7	0.9	1.1	3.8
FPR	**0.9**	10.4	-	.02	2	-	.76
Precision	**99**	-	-	-	-	98.95	-

V. CONCLUSION

An outlier detection system is proposed to identify the normal and abnormal network traffic. The SVM and GNB classification algorithm are employed to classify the behaviours of incoming network connections that affect a network of computers. They are built and evaluated on the NSL-KDD and CICIDS2017 network traffic datasets. Data mining preprocessing stages for network flow data, besides tuning the SVM's RBF control parameters and GNB's smoothing parameter, were vital for improving the inclusive effectiveness of the proposed NODS. The performance of the proposed system is compared with other related IDSs and the evaluation results stated the superiority of the proposed SVM-NODS in detecting the different intrusions. In our future work, we will explore and implement other strategies for boosting the detection system capabilities and also investigate many deep learning trend models in building the proposed detection model.

ACKNOWLEDGMENT

The authors would like to thank the University of Jeddah for its technical support.

REFERENCES

[1] J. Jang-Jaccard and S. Nepal, "A survey of emerging threats in cybersecurity," *J. Comput. Syst. Sci.*, vol. 80, no. 5, pp. 973–993, Aug. 2014.

[2] P. Schaik, J. Jansen, J. Onibokun, J. Camp, and P. Kusev, "Security and privacy in online social networking: Risk perceptions and precautionary behaviour," *Comput. Hum. Behav.*, vol. 78, pp. 283–297, Jan. 2018.

[3] K. Singh and S. Upadhyaya, "Outlier detection: Applications and techniques," *J. Netw. Comput. Appl.*, vol. 9, p. 307, Jan. 2012.

[4] J. Branch, C. Giannella, B. Szymanski, E. Wolff, and H. Kargupta, "In-network outlier detection in wireless sensor networks," *Knowl. Inf. Syst.*, vol. 34, pp. 23–54, Jan. 2013.

[5] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: Techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, pp. 19–38, Dec. 2019.

[6] J. Zhang and M. Zulkernine, "Anomaly based network intrusion detection with unsupervised outlier detection," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2006, pp. 2388–2393.

[7] P. Gogoi, D. K. Bhattacharyya, B. Borah, and J. K. Kalita, "A survey of outlier detection methods in network anomaly identification," *Comput. J.*, vol. 54, no. 4, pp. 570–588, Apr. 2011.

[8] S. Suthaharan, "Support vector machine," in *Machine Learning Models and Algorithms for Big Data Classification* (Integrated Series in Information Systems), vol. 36. Berlin, Germany: Springer, 2016, pp. 207–235.

[9] L. Dhanabal and S. P. Shantharajah, "A study on NSL-KDD dataset for intrusion detection system based on classification algorithms," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 4, no. 6, pp. 446–452, 2015.

[10] A. Gaddam, T. Wilkin, M. Angelova, and J. Gaddam, "Detecting sensor faults, anomalies and outliers in the Internet of Things: A survey on the challenges and solutions," *Electronics*, vol. 9, no. 3, p. 511, 2020, doi: 10.3390/electronics9030511.

[11] L. Wawrowski, M. Michalak, A. Białas, R. Kurianowicz, M. Sikora, M. Uchroński, and A. Kajzer, "Detecting anomalies and attacks in network traffic monitoring with classification methods and XAI-based explainability," *Proc. Comput. Sci.*, vol. 192, pp. 2259–2268, Jan. 2021.

[12] X. Chen, H. Kim, J. M. Aman, W. Chang, M. Lee, and J. Rexford, "Measuring TCP round-trip time in the data plane," in *Proc. Workshop Secure Program. Netw. Infrastructure*, Aug. 2020, pp. 35–41.

[13] J. Li, Y. Deng, W. Sun, W. Li, R. Li, Q. Li, and Z. Liu, "Resource orchestration of cloud-edge-based smart grid fault detection," *ACM Trans. Sensor Netw.*, vol. 18, no. 3, pp. 1–26, Aug. 2022, doi: 10.1145/3529509.

[14] S. S. Chakkaravarthy, D. Sangeetha, and V. Vaidehi, "A survey on malware analysis and mitigation techniques," *Comput. Sci. Rev.*, vol. 32, pp. 1–23, May 2019.

[15] M. Safaei Pour, C. Nader, K. Friday, and E. Bou-Harb, "A comprehensive survey of recent internet measurement techniques for cyber security," *Comput. Secur.*, vol. 128, May 2023, Art. no. 103123.

[16] H. Lu, Y. Zhu, M. Yin, G. Yin, and L. Xie, "Multimodal fusion convolutional neural network with cross-attention mechanism for internal defect detection of magnetic tile," *IEEE Access*, vol. 10, pp. 60876–60886, 2022.

[17] S. Wang, X. Hu, J. Sun, and J. Liu, "Hyperspectral anomaly detection using ensemble and robust collaborative representation," *Inf. Sci.*, vol. 624, pp. 748–760, May 2023.

[18] Z. Wu, J. Cao, Y. Wang, Y. Wang, L. Zhang, and J. Wu, "HPSD: A hybrid PU-learning-based spammer detection model for product reviews," *IEEE Trans. Cybern.*, vol. 50, no. 4, pp. 1595–1606, Apr. 2020, doi: 10.1109/TCYB.2018.2877161.

[19] Y. Yao, J. Zhao, Z. Li, X. Cheng, and L. Wu, "Jamming and eavesdropping defense scheme based on deep reinforcement learning in autonomous vehicle networks," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 1211–1224, 2023, doi: 10.1109/TIFS.2023.3236788.

[20] X. Xie, L. Huang, S. M. Marson, and G. Wei, "Emergency response process for sudden rainstorm and flooding: Scenario deduction and Bayesian network analysis using evidence theory and knowledge meta-theory," *Natural Hazards*, vol. 117, no. 3, pp. 3307–3329, Jul. 2023, doi: 10.1007/s11069-023-05988-x.

[21] B. Cao, J. Zhao, Y. Gu, Y. Ling, and X. Ma, "Applying graph-based differential grouping for multiobjective large-scale optimization," *Swarm Evol. Comput.*, vol. 53, Mar. 2020, Art. no. 100626, doi: 10.1016/j.swevo.2019.100626.

[22] M. A. Manzoor and Y. Morgan, "Network intrusion detection system using apache storm," *Adv. Sci., Technol. Eng. Syst. J.*, vol. 2, no. 3, pp. 812–818, Jun. 2017.

[23] S.-J. Han and S.-B. Cho, "Evolutionary neural networks for anomaly detection based on the behavior of a program," *IEEE Trans. Syst., Man, Cybern., B*, vol. 36, no. 3, pp. 559–570, Jun. 2006.

[24] X. Wang, C. Zhang, and K. Zheng, "Intrusion detection algorithm based on density, cluster centers, and nearest neighbors," *China Commun.*, vol. 13, no. 7, pp. 24–31, Jul. 2016.

[25] K. S. Desale, C. N. Kumathekar, and A. P. Chavan, "Efficient intrusion detection system using stream data mining classification technique," in *Proc. Int. Conf. Comput. Commun. Control Autom.*, Feb. 2015, pp. 469–473.

[26] J. Zhang, H. Li, Q. Gao, H. Wang, and Y. Luo, "Detecting anomalies from big network traffic data using an adaptive detection approach," *Inf. Sci.*, vol. 318, pp. 91–110, Oct. 2015.

[27] D. Stiawan, M. Y. B. Idris, A. M. Bamhdi, and R. Budiarto, "CICIDS-2017 dataset feature analysis with information gain for anomaly detection," *IEEE Access*, vol. 8, pp. 132911–132921, 2020.

[28] P. Jairu and A. B. Mailewa, "Network anomaly uncovering on CICIDS-2017 dataset: A supervised artificial intelligence approach," in *Proc. IEEE Int. Conf. Electro Inf. Technol. (eIT)*, Mankato, MN, USA, May 2022, pp. 606–615.

[29] S. Patil, V. Varadarajan, S. M. Mazhar, A. Sahibzada, N. Ahmed, O. Sinha, S. Kumar, K. Shaw, and K. Kotecha, "Explainable artificial intelligence for intrusion detection system," *Electronics*, vol. 11, no. 19, p. 3079, 2022.

[30] O. Elnakib, E. Shaaban, M. Mahmoud, and K. Emara, "EIDM: Deep learning model for IoT intrusion detection systems," *J. Supercomput.*, vol. 79, no. 12, pp. 13241–13261, Aug. 2023.

[31] Y. Hou, S. G. Teo, Z. Chen, M. Wu, C.-K. Kwoh, and T. Truong-Huu, "Handling labeled data insufficiency: Semi-supervised learning with self-training mixup decision tree for classification of network attacking traffic," *IEEE Trans. Dependable Secure Comput.*, early access, Aug. 1, 2022, doi: 10.1109/TDSC.2022.3195534.

[32] A. Smiti, "A critical overview of outlier detection methods," *Comput. Sci. Rev.*, vol. 38, Nov. 2020, Art. no. 100306.

[33] B. Deka, "Pattern recognition and machine intelligence," in *Proc. 8th Int. Conf.*, Tezpur, India. Cham, Switzerland: Springer, Dec. 2019, pp. 56–64.

[34] R. Aliakbarisani, A. Ghasemi, and S. Felix Wu, "A data-driven metric learning-based scheme for unsupervised network anomaly detection," *Comput. Electr. Eng.*, vol. 73, pp. 71–83, Jan. 2019, doi: 10.1016/j.compeleceng.2018.11.003.

[35] W. Bul'ajoul, A. James, and M. Pannu, "Improving network intrusion detection system performance through quality of service configuration and parallel technology," *J. Comput. Syst. Sci.*, vol. 81, no. 6, pp. 981–999, Sep. 2015.

[36] P. V. Alvarado, "Design of a traffic generation platform for offline evaluation of NIDS," in *Proc. 8th Int. Conf. Adv. Comput. Control Netw. ACCN*, Jun. 2018, pp. 11–15.

[37] I. Sumaiya Thaseen and C. Aswani Kumar, "Intrusion detection model using fusion of chi-square feature selection and multi class SVM," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 29, no. 4, pp. 462–472, Oct. 2017.

[38] S. S. Sathiyadhas and M. C. V. Soosai Antony, "A network intrusion detection system in cloud computing environment using dragonfly improved invasive weed optimization integrated shepard convolutional neural network," *Int. J. Adapt. Control Signal Process.*, vol. 36, no. 5, pp. 1060–1076, May 2022.

[39] S. Panwar and Y. Raiwani, "Data reduction techniques to analyze NSL-KDD dataset," *Int. J. Comput. Eng. Technol*, vol. 5, no. 10, pp. 21–31, 2017.

[40] K. Keerthi Vasan and B. Surendiran, "Dimensionality reduction using principal component analysis for network intrusion detection," *Perspect. Sci.*, vol. 8, pp. 510–512, Sep. 2016.

[41] M. A. Hall and L. A. Smith, "Feature selection for machine learning: Comparing a correlation-based filter approach to the wrapper," in *Proc. 12th Int. FLAIRS Conf.*, 1999, pp. 235–239.

[42] M. Hosseinzadeh, A. M. Rahmani, B. Vo, M. Bidaki, M. Masdari, and M. Zangakani, "Improving security using SVM-based anomaly detection: Issues and challenges," *Soft Comput.*, vol. 25, no. 4, pp. 3195–3223, 3195.

[43] S. A. Ajila and A. A. Bankole, "Using machine learning algorithms for cloud client prediction models in a web VM resource provisioning environment," *Trans. Mach. Learn. Artif. Intell.*, vol. 4, no. 1, p. 28, Feb. 2016, doi: 10.14738/tmlai.41.1690.

[44] R. Duo, X. Nie, N. Yang, C. Yue, and Y. Wang, "Anomaly detection and attack classification for train real-time Ethernet," *IEEE Access*, vol. 9, pp. 22528–22541, 2021, doi: 10.1109/access.2021.3055209.

[45] B. Zhang, Z. Liu, Y. Jia, J. Ren, and X. Zhao, "Network intrusion detection method based on PCA and Bayes algorithm," *Secur. Commun. Netw.*, vol. 2018, pp. 1–11, Nov. 2018.

[46] K. Bong and J. Kim, "Analysis of intrusion detection performance by smoothing factor of Gaussian NB model using modified NSL-KDD dataset," in *Proc. 13th Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, 2022, pp. 1471–1476.

[47] J. Wainer and P. Fonseca, "How to tune the RBF SVM hyperparameters? An empirical evaluation of 18 search algorithms," *Artif. Intell. Rev.*, vol. 54, no. 6, pp. 4771–4797, May 2021, doi: 10.1007/s10462-021-10011-5.

[48] A. H. Hamamoto, L. F. Carvalho, L. D. H. Sampaio, T. Abr ao, and M. L. Proença, "Network anomaly detection system using genetic algorithm and fuzzy logic," *Exp. Syst. Appl.*, vol. 92, pp. 390–402, Feb. 2018, doi: 10.1016/j.eswa.2017.09.013.

[49] Z. Chiba, "New anomaly network intrusion detection system in cloud environment based on optimized back propagation neural network using improved genetic algorithm," *Int. J. Commun. Netw. Inf. Secur. (IJCNIS)*, vol. 11, no. 1, pp. 61–84, Apr. 2022, doi: 10.17762/ijcnis.v11i1.3764.

[50] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," *Neurocomputing*, vol. 408, pp. 189–215, Sep. 2020.

[51] M. Tabassum, "A genetic algorithm analysis towards optimization solutions," *Int. J. Digit. Inf. Wireless Commun.*, vol. 4, no. 1, pp. 124–142, 2014.

[52] T. Alam, S. Qamar, A. Dixit, and M. Benaida, "Genetic algorithm: Reviews, implementations, and applications," *Int. J. Eng. Pedagogy (iJEP)*, vol. 10, no. 6, p. 57, Dec. 2020.

[53] (2009). *NSL-KDD Dataset for Network-based Intrusion Detection Systems*. [Online]. Available: http://nsl.cs.unb.ca/KDD/NSLKDD.html

[54] S. S. Panwar, P. S. Negi, and Y. P. Raiwani, "Implementation of machine learning algorithms on CICIDS-2017 dataset for intrusion detection using WEKA," *Int. J. Recent Technol. Eng. (IJRTE)*, vol. 8, no. 3, pp. 2195–2207, Sep. 2019.

[55] R. Panigrahi and S. Borah, "A detailed analysis of CICIDS2017 dataset for designing intrusion detection systems," *Int. J. Eng. Technol.*, vol. 7, no. 3, pp. 479–482, 2018.

[56] M. E. Elhamahmy, H. N. Elmahdy, and I. A. Saroit, "A new approach for evaluating intrusion detection system," *Artif. Intell. Syst. Mach. Learn.*, vol. 2, pp. 290–298, Nov. 2010.

[57] P. Kar, S. Banerjee, K. C. Mondal, G. Mahapatra, and S. Chattopadhyay, "A hybrid intrusion detection system for hierarchical filtration of anomalies," in *Information and Communication Technology for Intelligent Systems*. Berlin, Germany: Springer, 2019, pp. 417–426.

[58] Y.-F. Hsu, Z. He, Y. Tarutani, and M. Matsuoka, "Toward an online network intrusion detection system based on ensemble learning," in *Proc. IEEE 12th Int. Conf. Cloud Comput. (CLOUD)*, Jul. 2019, pp. 174–178.

[59] S. Sarvari, N. F. Mohd Sani, Z. Mohd Hanapi, and M. T. Abdullah, "An efficient anomaly intrusion detection method with feature selection and evolutionary neural network," *IEEE Access*, vol. 8, pp. 70651–70663, 2020.

[60] A. Alsaleh and W. Binsaeedan, "The influence of salp swarm algorithm-based feature selection on network anomaly intrusion detection," *IEEE Access*, vol. 9, pp. 112466–112477, 2021.

[61] L. Almuqren, M. S. Maashi, M. Alamgeer, H. Mohsen, M. A. Hamza, and A. A. Abdelmageed, "Explainable artificial intelligence enabled intrusion detection technique for secure cyber-physical systems," *Appl. Sci.*, vol. 13, no. 5, p. 3081, Feb. 2023.

[62] Y. N. Rao and K. S. Babu, "An imbalanced generative adversarial network-based approach for network intrusion detection in an imbalanced dataset," *Sensors*, vol. 23, no. 1, p. 550, Jan. 2023.

OMAR ALGHUSHAIRY received the bachelor's degree in information systems from King Abdulaziz University, Saudi Arabia, the master's degree in computer science from the University of Bridgeport, USA, and the Ph.D. degree in computer science from the University of Idaho, USA. He is currently an Assistant Professor in data science and AI with the College of Computer Science and Engineering, University of Jeddah, Saudi Arabia. His research interests include outlier detection, data stream mining, machine learning, AI, and evolutionary computation.

RAED ALSINI received the M.Sc. degree in computer science from California State University, Fullerton, USA, in 2016, and the Ph.D. degree in computer science from the University of Idaho, USA, in 2021. He is currently an Assistant Professor with the Information Systems Department, Faculty of Computing and Information Technology, King Abdulaziz University. His research interests include big data, anomaly detection, streaming data, artificial intelligence, and cybersecurity.

ZAKHRIYA ALHASSAN received the Ph.D. degree from the College of Computer Sciences, Durham University, Durham, U.K., in 2021. He is currently an Assistant Professor with the College of Computer Science and Engineering, University of Jeddah, Jeddah, Saudi Arabia. Before joining Durham, he was with Saudi Aramco, General Electric (GE) and Saudi Electricity Company (SEC), Saudi Arabia, in the area of business intelligence. His current research interests include business intelligence, medical data mining, clinical informatics, machine learning, and artificial networks.

ABDULRAHMAN A. ALSHDADI received the Ph.D. degree in cloud computing from the University of Southampton, Southampton, U.K., in February 2018. He is currently an Associate Professor in computer science with the College of Computer Science and Engineering, University of Jeddah. His research interests include industry 4.0 prestaining issues of cloud computing and fog computing security, the Internet of Things (IoT) and smart cities, intelligent systems, deep learning, data science analytics, and modelling. He has published numerous conference papers, journal articles, and one book chapter.

AMEEN BANJAR received the Ph.D. degree in distributed network functions virtualization from the University of Technology, Sydney, Australia, in November 2016. He is currently an Associate Professor in information technology telecommunication with the College of Computer Science and Engineering (CCSE), University of Jeddah. His research interests include industry 4.0, involving intelligent digital technology, machine learning and deep learning to create a more holistic and better-connected ecosystem of companies focusing on manufacturing, data science analytics, and modelling. He has published numerous conference papers, journal articles, and book chapters.

AYMAN YAFOZ received the M.Sc. degree in web technology from the University of Southampton, U.K., in 2015, and the Ph.D. degree in computer science from the University of Regina, Canada, in 2021. He is currently an Assistant Professor with the Information Systems Department, Faculty of Computing and Information Technology, King Abdulaziz University. His research interests include natural language processing and data science.

XIAOGANG (MARSHALL) MA received the Ph.D. degree in earth systems science and GIScience from the University of Twente, The Netherlands, in 2011. He is currently an Associate Professor in computer science with the University of Idaho. Then, he completed his postdoctoral training in data science with the Rensselaer Polytechnic Institute. His research interests include deploying data science in the semantic web to support cross-disciplinary collaboration and scientific discovery, with broad interests in complex systems in earth and environmental sciences, data interoperability and provenance, and visualized exploratory analysis of big and small data. He was one of the four invited early-career panelists at the 2016 International Data Week. He is active in international societies of data science and geoinformatics, including ACM SIGWEB, CODATA, ESIP, RDA, GSA, AGU, and IAMG. He received the Science of Team Science (SciTS) Meritorious Contribution Award, in 2018, the IAMG A.B. Vistelius Research Award, in 2015, and the inaugural ICSU-WDS Data Stewardship Award, in 2014.