Hybrid Optimal Feature Subset Classification for Improved Alert Reduction in Intrusion Detection Systems

E.N. Osuigbo, U. A. Okengwu

Abstract— Intrusion detection systems (IDS) are recently deployed as one of the major network security components for a safe network, especially in large corporations. Events in a network are monitored and analysed by multiple IDSs in order to detect intrusions. An IDS connected to a network generates a large volume of alerts some are real or true alerts and many others are false positive or irrelevant alerts. Network Analysts have continuously encountered challenges caused by the difficulty in identifying threats and taking immediate remedial actions due to the abundance of false alerts. Recent studies have used machine learning algorithms to solve this problem by detecting the most characteristic threats to Intrusion Detection Systems. In this work, we propose an improved model using a hybrid machine learning approach to determine the most serious alerts and reduce false alerts. Principal Component Analysis (PCA) is used for feature selection combined with Genetic Algorithm (GA) optimizer to generate optimal feature subset in order to gain high Detection Rate (DR) with a significant reduction in False Alert Rate (FAR). The optimal feature subset is evaluated and classified using Support Vector Machine (SVM) algorithm. Results of the experiment and test on the UNSW-NB15 dataset showed that in comparison to other approaches, the proposed method can reach high detection accuracy and low false positive rate simultaneously. Our work will be of benefit to cooperate networked organizations, as it will ensure that only true alerts will be generated in such a way that security measures can be quickly implemented to secure the network from both internal and external attacks.

Index Terms— Alert Reduction, Cybersecurity, Decision Tree, Genetic Algorithm, Intrusion Detection, Principal Component Analysis, Support Vector Machine.

1 Introduction

Cybersecurity deals with the strategic defense of computer system and networks from the numerous danger and attacks that can cause security breaches and data theft or having other adverse effects to the computer systems and networks [23]. These threats are inevitable, even though one is cautious, there is no guarantee it will not happen. Any action that attempts to compromise the truthfulness, privacy or accessibility of an information is an Intrusion [3]. To manage such actions, an Intrusion Detection System (IDS) is deployed to observe network movement or system records for suspicious activities and informs the system or network administrator of danger by generating Alerts [10],[21],[22].

The core task of an IDS is to protect a computer system or computer network by identifying unfriendly occurrences on a network system or host device, observing the proceedings going on in a computer system or network and evaluating them for signs of intrusions [26]. An IDS is a security system that observes network traffic and computer systems. It works to investigate that traffic for possible unfriendly attacks are emanating from outside the organization and for misuse of system or attacks initiated from within the organization [12]. As an advancement, IDS enhance the network security and protects the data of the establishment. IDS can detect records and state any possible security occurrences as alerts [14]. These alerts notify the Network Administrator to get the data protected by taking necessary actions in contradiction of promising attacks. It can be either a software or hardware scheme to automate the intrusion detection procedures and is usually arranged inline, at a spanning port of a switch, or on a hub in place of a switch [1]. A perfect intrusion system must be capable of tolerating errors, which implies that it must stay alive despite system breakdown and not have its database restructured from the beginning [15].

Large organizations deploy multiple IDSs in their network infrastructure to monitor activities and keep their network safe [19],[2]. These IDSs observe network packets to detect unpleasant activities and generates an alert once such activities are detected to inform the network administrators enabling them to carryout quick remedial actions against any such danger. The alerts generated by IDSs are huge and can be categorized as true alerts and false positive alerts [11],[17]. The false alerts generate a very severe problem to intrusion detection systems as the large quantity of false positive alerts makes it problematic for the security administrator to recognise successful attacks and to carry out quick remedial schedules [8]. A robust model is required to remove irrelevant IDS alerts and generate only relevant alerts to reduce the number of alerts. This paper solves this problem by proposing a hybrid model combining feature selection, optimization and classification machine learning techniques to improved alert optimization and classification for reduction in Intrusion Detection Systems.

2 Related Works

Shittu et al. [18] suggested a framework named A Comprehensive System for Analysing Intrusion Alerts (ACSAnIA) for IDS alert reduction. Its post-correlation scheme comprised of a new prioritization metric centered on anomaly detection and a novel method to clustering events by means of correlation knowledge. The post-correlation scheme of ACSAnIA was appraised using data from a 2012 cyber range investigation carried out by an organization. Result shows that false-positives were magnificently minimized by 97%.

Al-Yaseen et al. [25] recommended a multi-level model for intrusion detection that combines the two techniques of modified K-means and support vector machine (SVM). Modified K-means is used to reduce the number of instances in a training data set and to construct new training data sets with high-quality instances. The new, high-quality training data sets are then utilized to train SVM classifiers. Consequently, the multi-level SVMs are employed to classify the testing data sets with high performance. The well-known KDD Cup 1999 data set is used to evaluate the proposed system.

Faraji and Abbaspour [4] proposed an online model for alert correlation. The model consists of two modules: (1) the online fuzzy clustering module which clusters alerts into fuzzy events based on their similarity and historical relevance; (2) the fuzzy inter event pattern mining which provides the first module with the historical relevance of alerts by mining frequent fuzzy patterns among them. Using these two modules, our approach is as fast as similarity-based approaches suitable for online alert correlation while it is able to extract complex attack scenarios like offline time-consuming data mining-based approaches. Furthermore, observing the frequent events makes our approach capable of detecting scenarios including wrapping tricks which tries to fake the source or destination IPs. The experimental results with the well-known dataset DARPA 2000 and the ISCX UNB intrusion detection evaluation dataset proved mentioned claims.

Athira and Pathari [2] opined that, generating abundant false alert has conveyed a severe assignment in Intrusion discovery systems. In their work, they improved on correlation process and alerts were organized from diverse intrusion detection systems like snort, ossec, suricata and were later, classified using machine learning technique, which aided the reasonable reduction of false positives. Major part of the work concentrates on the collection of alerts from different intrusion detection systems to represent them in IDMEF (Intrusion Detection Message Exchange Format) format.

Bostani and Sheikhan [6] proposed a novel real-time hybrid intrusion detection framework that consists of anomaly-based and specification-based intrusion detection modules for detecting two well-known routing attacks in IoT called sinkhole and selective-forwarding attacks. The specification-based intrusion detection agents, that are located in the router nodes, analyze the behavior of their host nodes and send their local results to the root node through normal data packets while an anomaly-based intrusion detection agent, that is located in the root node, employs the unsupervised optimum-path forest algorithm for projecting clustering models by using incoming data packets. This agent, which is based on the MapReduce architecture, can work in a distributed platform for projecting clustering models and consequently parallel detecting of anomalies as a global detection approach.

Shah et al. [24] applied Dempster–Shafer (DS) rule to achieve alert fusion, classification and removal of unwanted alerts. This recommended alert fusion technique bettered the performance of recognition by reducing alert excesses generated by high volume of false alerts. The removal of unwanted alerts was achieved due to the fact that the DS rule took responsibility of all confirmation bases to be reliable and combines variable consistency of IDS from conflict between true positive rate of IDS.

Aljawarneh et al. [20] developed a new hybrid model that can be used to estimate the intrusion scope threshold degree based on the network transaction data's optimal features that were made available for training utilizing the NSL-KDD data set, the binary and multiclass problem with a 20% testing dataset. The experimental results revealed that the hybrid approach had a significant effect on the minimisation of the computational and time complexity involved when determining the feature association impact scale.

Min et al. [5] proposed an autoencoder-based framework, i.e., SU-IDS, for semi-supervised and unsupervised network intrusion detection. The framework augments the usual clustering (or classification) loss with an auxiliary loss of autoencoder, and thus achieves a better performance. The experimental results on the classic NSL-KDD dataset and the modern CICIDS2017 dataset show the superiority of our proposed models.

Jiadong et al. [17] who proposed an effective IDS by using hybrid data optimization which consists of two parts: data sampling and feature selection, called DO_IDS. In data sampling, the Isolation Forest (iForest) is used to eliminate outliers, genetic algorithm (GA) to optimize the sampling ratio, and the Random Forest (RF) classifier as the evaluation criteria to obtain the optimal training dataset. In feature selection, GA and RF are used again to obtain the optimal feature subset. Finally, an intrusion detection system based on RF is built using the optimal training dataset obtained by data sampling and the features selected by feature selection. The model is taking a lot of time to train classifiers and it is recommended that the search strategy be further optimized.

Singh et al. [16] designed a soft-computing-based scheme was designed to minimize the false positive rate for data of anomaly type of intrusion detection system. A neural network model was adopted to categorize available datasets and normal occasions for several subclasses. The designed approach is nice as it does not have need of any knowledge of the dataset pattern. Evaluation is carried out on the numerous attacks available using the KDDCup'99 and NSL-KDD data sets. Result shows that this method is advantageous for real-life circumstances with a low existence of attacks and a reduced amount of false positive alert.

Sun et al. [7] proposed an alert aggregation scheme that is based on conditional rough entropy and knowledge granularity to solve the problem of repetitive and redundant alert information in network security devices. Firstly, we use conditional rough entropy and knowledge granularity to determine the attribute weights. This method determined the different important attributes and their weights for different types of attacks and calculates the similarity value of two alerts by weighting based on the results of attribute weighting. The sliding time window method is used to aggregate the alerts whose similarity value is larger than a threshold, which is set to reduce the redundant alerts. The proposed scheme is applied to the CIC-IDS 2018 dataset and the DARPA 98 dataset. The experimental results show that this method can effectively reduce the redundant alerts and improve the efficiency of data processing, thus providing accurate and concise data for the next stage of alert fusion and analysis.

Heigl et al. [9] introduced a novel framework called Streaming Outlier Analysis and Attack Pattern Recognition, denoted as SOAAPR, which is able to process the output of various online unsupervised OD methods in a streaming fashion to extract information about novel attack patterns. Three different privacy-preserving, fingerprint-like signatures are computed from the clustered set of correlated alerts by SOAAPR, which characterizes and represents the potential attack scenarios with respect to their communication relations, their manifestation in the data's features and their temporal behavior. Beyond the recognition of known attacks, comparing derived signatures, they can be leveraged to find similarities between yet unknown and novel attack patterns.

3 Design Methodology

The system is designed using a hybrid approach combining machine learning algorithms for feature selection, optimization and classification of alerts in intrusion detection systems.

3.1 HOAC_IDS Model

In the network, the normal behavior of users is more than the anomalous behavior, which makes the data distribution of normal behaviors and anomalous behaviors unbalanced. To enhance the detection performance of IDS, a robust Hybridized Alert Optimization and Classification model for IDS alert reduction here after referred to as HAOC_IDS. It tackles the problems of the existing system by combining novel machine learning algorithms, Principal Component Analysis (PCA), Genetic Algorithm (GA), and Support Vector Machine (SVM) to improve IDS alert assessment and reduction. PCA is used for data sampling and feature selection. GA is an optimization heuristic combining direct and stochastic search within a solution space, whereas SVM is used for solving classification jobs.

HAOC _IDS consists of three phases: feature selection, Optimization and Classification. In the feature selection phase, UNSW-NB15 dataset is preprocessed by transforming the symbolic valued attributes to numeric and applying the PCA algorithm. In the Optimization phase, Genetic Algorithm is employed for feature selection. SVM is used as the classifier engine in the classification phase. The SVM parameters are selected from the optimal feature set from the second phase.

PCA is used for dimensionality reduction to reduce the number of IDS alert variables that are correlated to each other into fewer independent IDS alert variables without losing the essence of these variables.

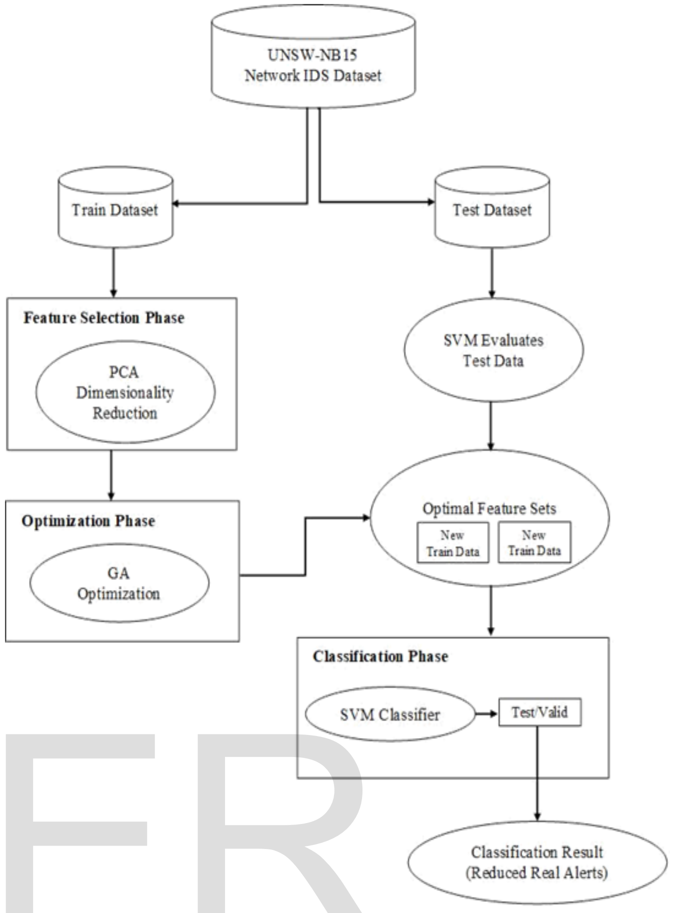


Fig. 1. Architecture of HAOC-IDS Model

GA is used as the optimization engine to optimize the sampling ratio and used to obtain the optimal feature subset. GA optimizer mainly comprised of three operators: selection, crossover, and mutation. The selection operator selects a good string on the basis of fitness to breed a new generation. Crossover operator has the responsibility of combining good strings to generate better offspring while mutation operator has the responsibility to alter a string locally to maintain genetic diversity from one generation of a population of chromosomes to the next. The additional reproduction component shows the emergence of a new generation from the whole genetic algorithm process that generates optimal set. The model uses real-valued GA, although its chromosome comprises of integer values as well. The chromosome consists of 3 genes, referred to as G1, G2 and G3 and Fitness function is also defined. Each Chromosome is characterized as a set of genes G = <G1;G2;G3> is represented. G1 is the non-negative integer value, that represents the algorithm used for classification. G2 is the real value, cost parameter C, and G3 is the real value, which represents bias term Accuracy has been carefully chosen for fitness evaluation, thus the quest to obtain a classifier with best accuracy performance.

Finally, the SVM classifier component performs classification of alerts into real and false alerts, minimizing alerts size by plotting the training vectors in high dimensional feature space through nonlinear mapping and labelling each vector by its class. The data is then classified by determining a set of support vectors, which are members of the set of training inputs that outline a hyperplane in the feature space. It maps pattern vectors to a high dimensional feature space to generate best separating hyperplane. The output from the SVM classifier component is sent to the network administrator for visualization. The proposed system is efficient to accomplish optimal detection, thus, reducing the overheads and maximizing performance of SVM classifier. The entire process as described above is shown in Figure 1. The algorithm for HAOC-IDS model is an order that shows the steps used for the assessment and reduction of alerts from multiple IDSs.

3.2 HOAC_IDS Algorithm

Algorithm 1: HOC-IDS

Step 1:

Initialize a real-valued γ population.

GA-SVM(n, c1, c2, rangeC, rangeBias, terminate_iterations, max_iterations)

γ ←3 (number of chromosome dimensions, representing different SVM classifiers as described ealier)

perf ← []

Step 2:

Evaluate the fitness of each γ. For each γ:

2.1 Initialize a real-valued C population.

```
$cl \leftarrow \{i \mid cl_{\min} \le i \le cl_{\max}, cl_{\min} \in Z, i \in Z, cl_{\max} \in Z\}$
$global\_fitness \leftarrow 0$
$term\_iterations \leftarrow 0$
$t \leftarrow 0$                      number of iterations
$P \leftarrow Init(n)$                Initialize a 3-dimensional chromosome genes
```

2.2 Evaluate the fitness of each C with the fixed γ, i.e.

a) Train SVM classifiers using cross validation with each C and the fixed γ.

```
for $\forall p_x \in P$
    $p_{x1} \leftarrow cl_{min} + round(rand(0,1) * (cl_{max} - cl_{min}))$
    $p_{x2} \leftarrow cl_{min} + rand(0,1) * (C_{max} - C_{min})$
    $p_{x3} \leftarrow b_{min} + rand(0,1) * (b_{max} - b_{min})$
    $y_p \leftarrow p;$
repeat
    if $no\_iterations = max\_iterations$ return SVM($y_p$);
    for $\forall p_x \in P$                      set the personal best position
        $f(x_p) \leftarrow evalSVM(p_{x1}, p_{x2}, p_{x3})$
        if $f(x_p) < \hat{y}(t)$                 set the global best position
            $y_p \leftarrow x_p;$
            $term\_iterations \leftarrow 1$      no need to terminate, continue searching
        else
            $term\_iterations \leftarrow term\_iterations + 1$
        if $f(y_p) < f(\hat{y})$   $\hat{y} = y_p$
```

b) Calculate the fitness corresponding to the accuracy rate of cross validation.

```
    for $\forall p_x \in P$
        for j=1:k
            $Vmax \leftarrow \delta_j \times (R_{max, j} - R_{min, j})$       Maximum allowed velocity
            if (j = 1)
                $Vmax \leftarrow round(Vmax);$
                $v_{pj}(t+1) = v_{pj}(t) + round(c_1 \times rand(0,1) \times (y_{pj}(t) - x_{pj}(t)) + c$
            else
                $v_{pj}(t+1) \leftarrow v_{pj}(t) + c_1 \times rand(0,1) \times (y_{pj}(t) - x_{pj}(t)) + c_2 \times rand(0,1) \times (\hat{y}_j(t) - x$
            $v_{pj}(t+1) \leftarrow (v_{pj}(t+1) < Vmax\ ?\ v_{pj}(t+1) : Vmax)$
        $x_p(t+1) \leftarrow x_p(t) + v_p(t+1)$
        $y_p(t+1) \leftarrow \begin{cases} y_p(t), if\ \ f(x_p(t+1)) \le f(y_p(y)) \\ y_p(t+1), if\ \ f(x_p(t+1)) > f(y_p(y)) \end{cases}$
```

2.3 Set the best fitness of the fixed γ with each C in C population as the fitness of the γ.

2.4 Go to Step 3 if termination criterion is satisfied, or go to Step 2.5 otherwise.

2.5 Evolve a new C population by crossover, mutation and selection. Go to Step 2.2.

Step 3:

Go to Step 5 if termination criterion is satisfied, or go to Step 4 otherwise.

Step 4:

Evolve a new γ population by crossover, mutation and selection. Go to Step 2.

Step 5:

Select the final {C, γ} corresponding to the best fitness (i.e., the highest accuracy of cross validation).

```
        if $x_{p1}(t+1) > cl_{mas}$
            $x_{p1}(t+1) \leftarrow cl_{min};$
        if  $x_{p2}(t+1) < C_{min})$
            $x_{p2}(t+1) \leftarrow C_{min};$
    $\hat{y}(t) \leftarrow \min(f(y_0(t)), \ldots, f(y_n(t)))$
    $t \leftarrow t+1$
until ($term\_iterations < terminate\_iterations$)
```

Step 6:

Output: Optimal linear classifier (DT (y_p), RF (y_p), SVM (y_p))

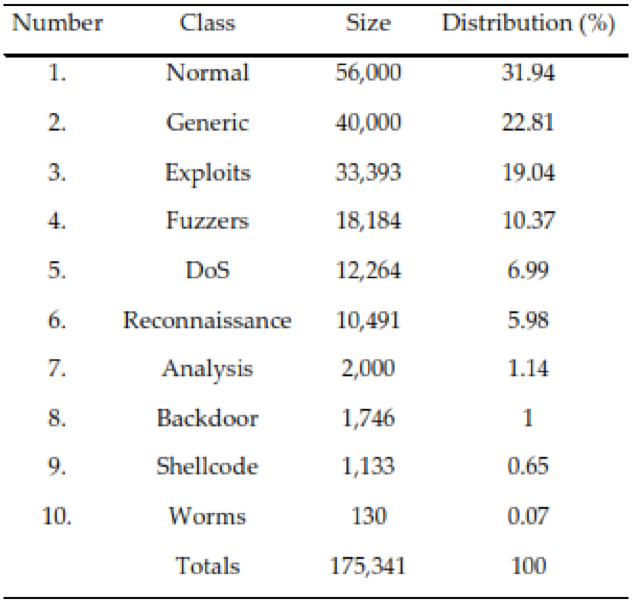
4 Experiment and Results

4.1 Experiment Setup

To analyse the classification effectiveness of HAOC_IDS model, an experiment was performed using UNSW-NB15 dataset. The parameters used in the algorithm are obtained by empirical value and set such that in feature selection and data sampling, considering the efficiency factor, the numbers of components of PCA are set as 10. In data optimization to generate optimal feature set, genetic algorithm values include population initiation N = 100, the crossover probability Pcrossover = 0.8, the mutation probability Pmutation = 0.1, and the termination condition (the number of descendants inherited) G = 50. In classifier training, the number of the SVM kernels is set as 200. The proposed technique was compared with the hybrid other uncombine approaches.

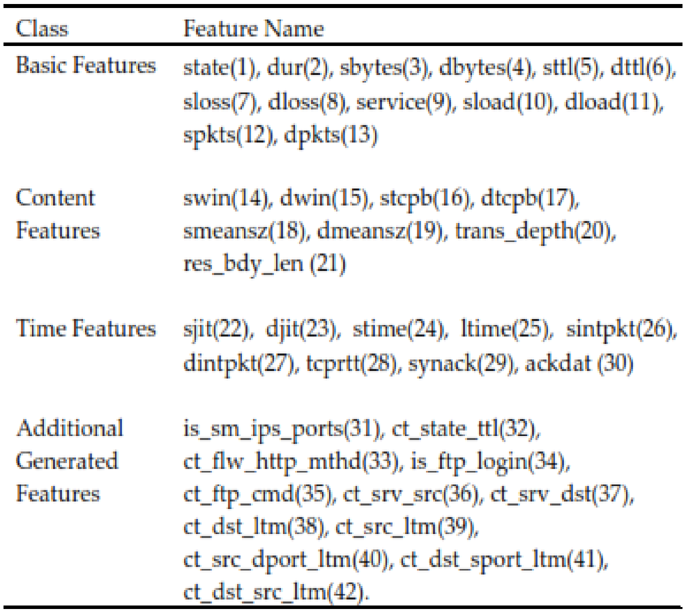
To perform the experiment, PyCharm Professional Edition 2021.3.1 environment was used on the version of Anaconda 2020.11. The need to develop software to simulate the working of HAOC_IDS model necessitated the choice of the conventional Object-Oriented paradigm (OOP) and hence programmed using Python programming language. A PC with Intel(R) Core (TM) i5-4460 at 3.6 GHz CPU, 8GB RAM, 1024 * 768 screen resolution is a minimal hardware required. The system is required to be running Window Operating System (7 and above), Linux Operating system with X windows capability or MAC OS (OS X 10.9 and above) with Python using Pycharm (2017 and above) and Anaconda (3 and above).

TABLE 1
UNSW-NB15 Parameters



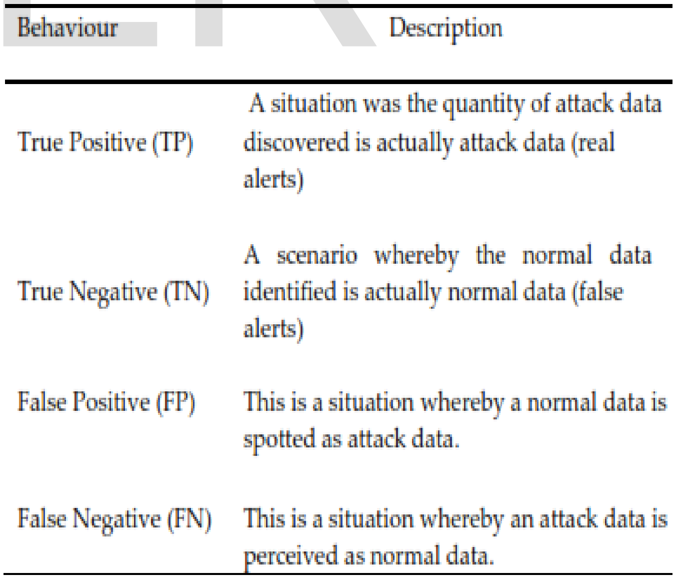
Number	Class	Size	Distribution (%)
1.	Normal	56,000	31.94
2.	Generic	40,000	22.81
3.	Exploits	33,393	19.04
4.	Fuzzers	18,184	10.37
5.	DoS	12,264	6.99
6.	Reconnaissance	10,491	5.98
7.	Analysis	2,000	1.14
8.	Backdoor	1,746	1
9.	Shellcode	1,133	0.65
10.	Worms	130	0.07
	Totals	175,341	100

TABLE 2
UNSW-NB15 Feature Set



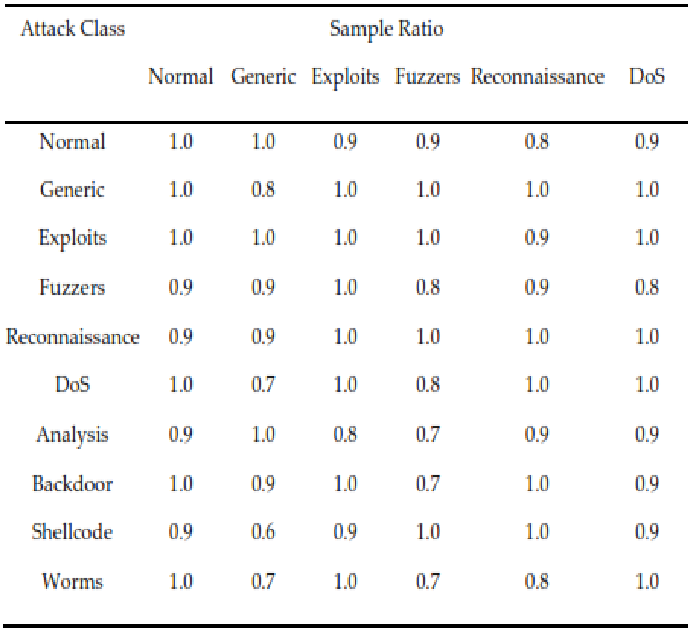
Class	Feature Name
Basic Features	state(1), dur(2), sbytes(3), dbytes(4), sttl(5), dttl(6), sloss(7), dloss(8), service(9), sload(10), dload(11), spkts(12), dpkts(13)
Content Features	swin(14), dwin(15), stcpb(16), dtcpb(17), smeansz(18), dmeansz(19), trans_depth(20), res_bdy_len (21)
Time Features	sjit(22), djit(23), stime(24), ltime(25), sintpkt(26), dintpkt(27), tcprtt(28), synack(29), ackdat (30)
Additional Generated Features	is_sm_ips_ports(31), ct_state_ttl(32), ct_flw_http_mthd(33), is_ftp_login(34), ct_ftp_cmd(35), ct_srv_src(36), ct_srv_dst(37), ct_dst_ltm(38), ct_src_ltm(39), ct_src_dport_ltm(40), ct_dst_sport_ltm(41), ct_dst_src_ltm(42).

TABLE 3
Description of Attacks and Normal Data Behaviour



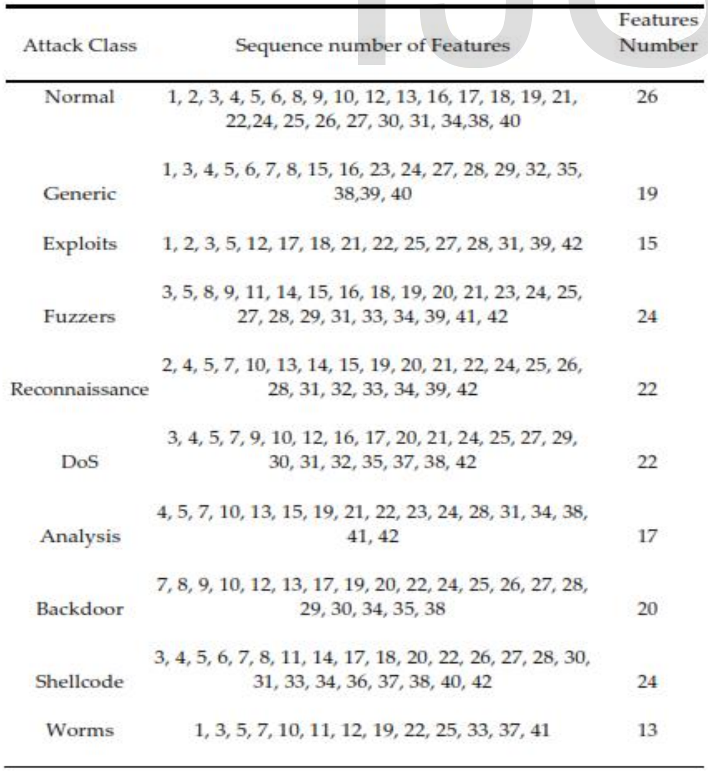
Behaviour	Description
True Positive (TP)	A situation was the quantity of attack data discovered is actually attack data (real alerts)
True Negative (TN)	A scenario whereby the normal data identified is actually normal data (false alerts)
False Positive (FP)	This is a situation whereby a normal data is spotted as attack data.
False Negative (FN)	This is a situation whereby an attack data is perceived as normal data.

TABLE 4
ATTACK CLASS WITH OPTIMAL SAMPLING RATIO



Attack Class	Sample Ratio					
	Normal	Generic	Exploits	Fuzzers	Reconnaissance	DoS
Normal	1.0	1.0	0.9	0.9	0.8	0.9
Generic	1.0	0.8	1.0	1.0	1.0	1.0
Exploits	1.0	1.0	1.0	1.0	0.9	1.0
Fuzzers	0.9	0.9	1.0	0.8	0.9	0.8
Reconnaissance	0.9	0.9	1.0	1.0	1.0	1.0
DoS	1.0	0.7	1.0	0.8	1.0	1.0
Analysis	0.9	1.0	0.8	0.7	0.9	0.9
Backdoor	1.0	0.9	1.0	0.7	1.0	0.9
Shellcode	0.9	0.6	0.9	1.0	1.0	0.9
Worms	1.0	0.7	1.0	0.7	0.8	1.0

TABLE 5
SELECTION OF OPTIMAL FEATURE SUBSET FOR EACH ATTACK CLASS



Attack Class	Sequence number of Features	Features Number
Normal	1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 13, 16, 17, 18, 19, 21, 22,24, 25, 26, 27, 30, 31, 34,38, 40	26
Generic	1, 3, 4, 5, 6, 7, 8, 15, 16, 23, 24, 27, 28, 29, 32, 35, 38,39, 40	19
Exploits	1, 2, 3, 5, 12, 17, 18, 21, 22, 25, 27, 28, 31, 39, 42	15
Fuzzers	3, 5, 8, 9, 11, 14, 15, 16, 18, 19, 20, 21, 23, 24, 25, 27, 28, 29, 31, 33, 34, 39, 41, 42	24
Reconnaissance	2, 4, 5, 7, 10, 13, 14, 15, 19, 20, 21, 22, 24, 25, 26, 28, 31, 32, 33, 34, 39, 42	22
DoS	3, 4, 5, 7, 9, 10, 12, 16, 17, 20, 21, 24, 25, 27, 29, 30, 31, 32, 35, 37, 38, 42	22
Analysis	4, 5, 7, 10, 13, 15, 19, 21, 22, 23, 24, 28, 31, 34, 38, 41, 42	17
Backdoor	7, 8, 9, 10, 12, 13, 17, 19, 20, 22, 24, 25, 26, 27, 28, 29, 30, 34, 35, 38	20
Shellcode	3, 4, 5, 6, 7, 8, 11, 14, 17, 18, 20, 22, 26, 27, 28, 30, 31, 33, 34, 36, 37, 38, 40, 42	24
Worms	1, 3, 5, 7, 10, 11, 12, 19, 22, 25, 33, 37, 41	13

4.2 UNSW-NB15 Dataset

The UNSW-NB15 dataset is created by the cyber security research group at the Australian Centre for Cyber Security (ACCS) recently [13]. The dataset contains 2, 540,044 records with 42 attributes, which is divided into training set and testing set. The training set contains 175,341 records, while the test set contains 82,332 records. The parameters of the dataset are shown in Table 1, and the feature description is shown in Table 2. Attacks and normal data behaviours are described in Table 3.

4.3 Results

The results of the experiment for the comparison of three algorithms is discussed in this section. The dataset contains both normal and attack data with different attack types consisting of Generic, Exploits, Fuzzers, DoS, Reconnaissance, Analysis, Backdoor, Shellcode, and Warms. The optimal sampling ratio of each class obtained during data sampling is shown in Table 4. Some of the attack class such as Analysis, Backdoor, Probe, and Worms are too small, so they are not considered for sampling.

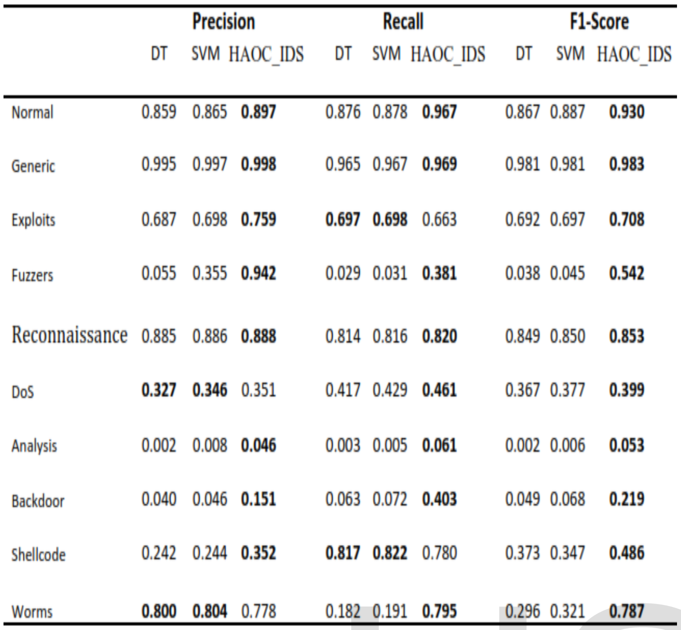
Table 5 shows the optimal feature subset of each attack class. It can be noted that Normal class has the largest number of features in the subset of optimal features, the number of its optimal features is 26, the least is Worms, and the number is 13. Compared with the total number of Features 42, all the classes have achieved considerable dimensionality reduction.

To verify the effectiveness of the HAOC_IDS hybridized model proposed in this work, we tested the model and obtained the precision, recall, and F1_score shown in the Table 6 and compared with the simple Decision Tree (DT) and Support Vector Machine (SVM) classifier without data sampling and feature selection. Results showed a slight decrease in the recall of Exploits and Shellcode, and the precision of Worms and DoS. Significant improvements are shown in other classes for the precision and recall, especially for class with less records, such as Analysis, Backdoor, Shellcode, and Worms. It can be seen from the results obtained that HAOC_IDS has achieved good performance on the detection of network anomaly behaviour with unbalanced data distribution.

Table 7 shows the comparison of detection accuracy and false alert rate (FAR) of all classes between simple DT, SVM and HAOC_IDS. FAR refers to the proportion of anomaly behaviours classified as normal to all anomaly behaviours. In the research of IDS, FAR is a significantly important evaluation indicator because in the network data, the number of normal behaviours is far more than the number of anomalous behaviours; even if all network data are classified as normal behaviour, the accuracy can reach a high level.

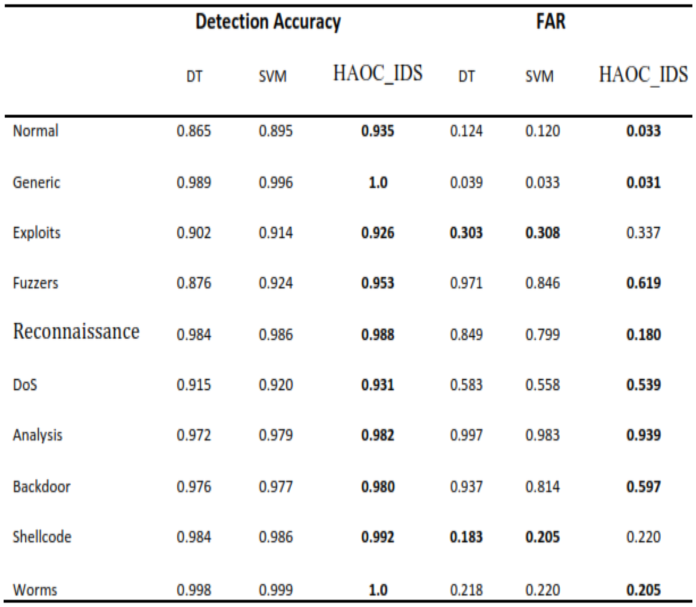
As seen from Table 7, both the simple models (DT and SVM) and HAOC_IDS have high classification accuracy in each class, but HAOC_IDS performed better than simple DT and SVM for FAR. From an integrated view, Table 8 shows the overall comparison of the whole dataset without specific class distinction between HAOC_IDS and simple DT and SVM without optimal feature subset.

TABLE 6
COMPARISON BETWEEN HAOC_IDS, DT AND SVM ON PRECISION, RECALL, AND F1 SCORE



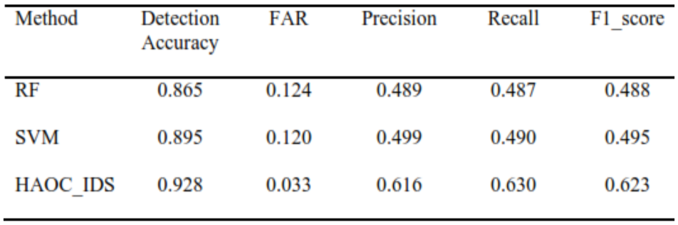
	Precision			Recall			F1-Score		
	DT	SVM	HAOC_IDS	DT	SVM	HAOC_IDS	DT	SVM	HAOC_IDS
Normal	0.859	0.865	**0.897**	0.876	0.878	**0.967**	0.867	0.887	**0.930**
Generic	0.995	0.997	**0.998**	0.965	0.967	**0.969**	0.981	0.981	**0.983**
Exploits	0.687	0.698	**0.759**	**0.697**	**0.698**	0.663	0.692	0.697	**0.708**
Fuzzers	0.055	0.355	**0.942**	0.029	0.031	**0.381**	0.038	0.045	**0.542**
Reconnaissance	0.885	0.886	**0.888**	0.814	0.816	**0.820**	0.849	0.850	**0.853**
DoS	**0.327**	**0.346**	0.351	0.417	0.429	**0.461**	0.367	0.377	**0.399**
Analysis	0.002	0.008	**0.046**	0.003	0.005	**0.061**	0.002	0.006	**0.053**
Backdoor	0.040	0.046	**0.151**	0.063	0.072	**0.403**	0.049	0.068	**0.219**
Shellcode	0.242	0.244	**0.352**	**0.817**	**0.822**	0.780	0.373	0.347	**0.486**
Worms	**0.800**	**0.804**	0.778	0.182	0.191	**0.795**	0.296	0.321	**0.787**

TABLE 7
COMPARISON BETWEEN HAOC_IDS, DT, AND SVM BASED ON DETECTION ACCURARCY AND FAR



	Detection Accuracy			FAR		
	DT	SVM	HAOC_IDS	DT	SVM	HAOC_IDS
Normal	0.865	0.895	**0.935**	0.124	0.120	**0.033**
Generic	0.989	0.996	**1.0**	0.039	0.033	**0.031**
Exploits	0.902	0.914	**0.926**	**0.303**	**0.308**	0.337
Fuzzers	0.876	0.924	**0.953**	0.971	0.846	**0.619**
Reconnaissance	0.984	0.986	**0.988**	0.849	0.799	**0.180**
DoS	0.915	0.920	**0.931**	0.583	0.558	**0.539**
Analysis	0.972	0.979	**0.982**	0.997	0.983	**0.939**
Backdoor	0.976	0.977	**0.980**	0.937	0.814	**0.597**
Shellcode	0.984	0.986	**0.992**	**0.183**	**0.205**	0.220
Worms	0.998	0.999	**1.0**	0.218	0.220	**0.205**

TABLE 8
OVERALL COMPARISON BETWEEN HOAC_IDS, DT AND SVM



Method	Detection Accuracy	FAR	Precision	Recall	F1_score
RF	0.865	0.124	0.489	0.487	0.488
SVM	0.895	0.120	0.499	0.490	0.495
HAOC_IDS	0.928	0.033	0.616	0.630	0.623

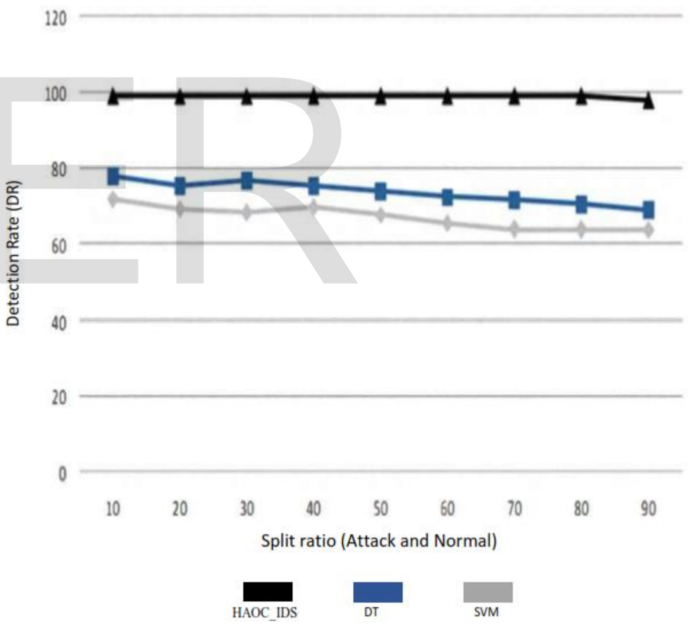


Fig. 2. Result showing comparison based on Detection Rate

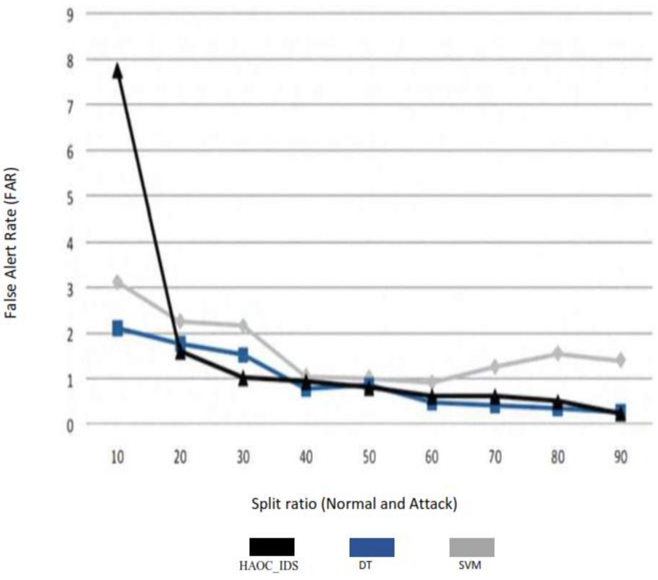


Fig. 3. Result showing comparison based on False Alert Rate

5 Conclusions

HAOC_IDS is an improved hybrid model designed to optimization and classify IDS alerts using a combination of feature selection (Principal Component Analysis), optimization (Genetic Algorithm) and classification (Support Vector Machine) techniques. An experiment was performed to determine the performance of the developed model. The findings of this study suggested that the three algorithms compared would likely misclassify an undesirably large amount of data when the number of attacks is higher than the number of normal data. We also found that FAR, HAOC_IDS performed better than others. However, DT and SVM showed increased detection rate. This also presents the efficiency of the developed model since good IDS should have a high detection rate as well as low false alert rate. From the test experiment on UNSW-NB15 dataset, results showed that in comparison to other models the proposed method can reach high detection accuracy and low false positive rate simultaneously. The combination of PCA, GA, and SVM for alert reduction can analyse large number of alerts from several IDSs and relate alerts to build a big picture of the occurrences, thus giving a high-level opinion of the security status. The optimized SVM algorithm has an outstanding benefit while applying it in multidimensional classification purpose, this can further reduce the influence of false positives and false negatives from numerous IDSs. The HAOC_IDS model used for IDS alert reduction is capable of figuring out optimal detection. This will reduce the overheads and increase performance of SVM classifier. HAOC_IDS model shows significant better results on false alarm rate and an efficient performance on detection rate. Thus, the proposed model is an effective technique for alert reduction.

Further research can explore a better HAOC_IDS parallelization, though this may increase the complexity of the overall infrastructure. For a more efficient calculation and caching of genetic kernels as well as the SVM classifier, new techniques should be investigated. Techniques like the Particle Swarm Optimization (PSO) neighbourhood and topology can be explored; thus, it also leaves room for improvement.

Acknowledgment

The authors wish to thank A, B, C. This work was supported in part by a grant from XYZ.

References

[1] A. Azordi, F. Cheng, and C. Meinel, “Towards Better Attack Path Visualizations Based on Deep Normalization of Host/Network IDS Alerts,” Proc. IEEE 30th International Conference on Advanced Information Networking and Applications (AINA ‘16), pp. 1064-1071, 2016.

[2] A.B. Athira and V. Pathari, “Standardisation and Classification of Alerts Generated by Intrusion Detection Systems,” International Journal on Cybernetics & Informatics, vol. 5, no. 2, 2016.

[3] C. Kafol, and A. Bregar, Cyber Security—Building a Sustainable Protection. DAAAM International Scientific Book, pp. 81-90.

[4] D.F. Faraji and M. Abbaspour, “Extracting Fuzzy Attack Patterns Using an Online Fuzzy Adaptive Alert Correlation Framework,” Security and Communication Networks, vol. 9, no. 14, 2016.

[5] E. Min, J. Long, Q. Liu, J. Cui, Z. Cai, and J. Ma, “Su-ids: A Semi-Supervised and Unsupervised Framework for Network Intrusion Detection,” Proc. International Conference on Cloud Computing and Security, pp. 322-334, 2018.

[6] H. Bostani and M. Sheikhan, “Hybrid of Anomaly-based and Specification-based IDS for Internet of Things using Unsupervised OPF Based on MapReduce Approach,” Computer Communications, vol. 98, pp. 52-71, 2017.

[7] J. Sun, L. Gu, and K. Chen, “An Efficient Alert Aggregation Method Based on Conditional Rough Entropy and Knowledge Granularity,” Entropy, vol. 22, no. 3, 2020.

[8] K. Goeschel, “Reducing False Positives in Intrusion Detection Systems using Data-Mining Techniques Utilizing Support Vector Machines, Decision Trees, and Naive Bayes for Off-Line Analysis,” Proc. IEEE Southeast Conference, pp. 1-6, 2016.

[9] M. Heigl, E. Weigelt, A. Urmann, D. Fiala, and M. Schramm, “Exploiting the Outcome of Outlier Detection for Novel Attack Pattern Recognition on Streaming Data,” Electronics, vol. 10, no. 17, 2021.

[10] M. Kumar and M. Hanumanthappa (2015). Cloud Based Intrusion Detection Architecture for Smartphones,” Proc. IEEE International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS ‘15), pp. 1-6. 2015.

[11] M. Lezzi, M. Lazoi, and A. Corallo, “Cybersecurity for Industry 4.0 in the Current Literature: A Reference Framework,” Computers in Industry, vol. 103, pp. 97-110, 2018.

[12] M.Z. Gunduz and R. Das, “Cyber-security on Smart Grid: Threats and Potential Solutions,” Computer networks, vol. 169, 2020.

[13] N. Moustafa and J. Slay, “The Evaluation of Network Anomaly Detection Systems: Statistical Analysis of the UNSW-NB15 Dataset and

the Comparison with the KDD99 Dataset," Information Security Journal: A Global Perspective, vol. 25, no. 3, pp. 18-31, 2016.

[14] P. Abhishek, B. Harsha, and S. Vaibhav, "Classification of Intrusion Detection System," International Journal of Computer Applications, vol. 118, no. 7, pp. 23-26, 2015.

[15] P. Hao, L. Zhe, S. Jie, L. Xue, C. Hanteng, L. Jianxin, and L. Lu, "Eagle: An Agile Approach to Automaton Updating in Cloud Security Services," Proc. IEEE Symposium on Service-Oriented System Engineering, pp. 73-80, 2016.

[16] P. Singh, S. Krishnamoorthy, A. Nayyar, A. K. Lunch, and A. Kaur, "Soft-Computing-Based False Alarm Reduction for Hierarchical Data of Intrusion Detection System," International Journal of Distributed Sensor Networks, vol. 15, no. 10, 2019.

[17] R. Jiadong, G. Jiawei, Q. Wang, Y. Huang, H. Xiaobing, and J. Hu, "Building an Effective Intrusion Detection System by Using Hybrid Data Optimization Based on Machine Learning Algorithms," Security and Communication Networks, pp. 11-22, 2019.

[18] R. Shittu, A. Healing, R. Ghanea-Hercock, R. Bloomfield, and M. Rajarajan, "Intrusion Alert Prioritisation and Attack Detection Using Post-Correlation Analysis," Computers & Security, vol. 50, pp. 1-15, 2015.

[19] R. Zuech, T.M. Khoshgoftaar, and R. Wald, "Intrusion Detection and Big Heterogeneous Data: A Survey," Journal of Big Data, vol. 2, no. 1, pp. 1-41, 2015.

[20] S. Aljawarneh, M. Aldwairi, and M.B. Yassein, "Anomaly-Based Intrusion Detection System through Feature Selection Analysis and Building Hybrid Efficient Model," Journal of Computational Science, vol. 25, pp. 152-160, 2018.

[21] S. Kumar and K. Dutta, "Intrusion detection in mobile ad hoc networks: techniques, systems, and future challenges. Security and Communication Networks, vol. 9, no. 14, pp. 2484-2556, 2016.

[22] S. Shamshirband, M. Fathi, A.T. Chronopoulos, A. Montieri, F. Palumbo, and A. Pescapè, "Computational Intelligence Intrusion Detection Techniques in Mobile Cloud Computing Environments: Review, Taxonomy, and Open Research Issues," Journal of Information Security and Applications, vol. 55, 2020.

[23] T. Hamid, D. Al-Jumeily, A. Hussain, and J. Mustafina, "Cyber Security Risk Evaluation Research Based on Entropy Weight Method.," Proc. IEEE International Conference on Developments in eSystems Engineering (DeSE), pp. 98-104, 2016.

[24] V. Shah, A. Aggarwal, and N. Chaubey, "Alert Fusion of Intrusion Detection Systems using Fuzzy Dempster Shafer Theory," Journal of Engineering Science and Technology Review, vol. 10, no. 3, pp. 123-127, 2017.

[25] W.L. Al-Yaseen, Z.A. Othman, and M.Z.A. Nazri, "Intrusion Detection System Based on Modified K-Means and Multi-level Support Vector Machines," International Conference on Soft Computing in Data Science, pp. 265-274, 2015.

[26] Z. El Mrabet, N. Kaabouch, and H. El-Ghazi, "Cyber-security in Smart Grid: Survey and Challenges. Computers & Electrical Engineering, vol. 67, pp. 469-482, 2018.