IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 31 May 2023, accepted 20 June 2023, date of publication 26 June 2023, date of current version 29 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3289405

RESEARCH ARTICLE

An Improved Design for a Cloud Intrusion Detection System Using Hybrid Features Selection Approach With ML Classifier

MHAMAD BAKRO[1], RAKESH RANJAN KUMAR[1], AMERAH ALABRAH[2], ZUBAIR ASHRAF[3], MD NADEEM AHMED[4], MOHAMMAD SHAMEEM[5], AND AHMED ABDELSALAM[6]

[1]Department of Computer Science and Engineering, C. V. Raman Global University, Bhubaneswar, Odisha 752054, India
[2]Department of Information Systems, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia
[3]Department of Computer Engineering and Applications, GLA University, Mathura, Uttar Pradesh 281406, India
[4]Department of AIT-Computer Science Engineering (CSE), Chandigarh University, Chandigarh, Punjab 140413, India
[5]Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Guntur, Andhra Pradesh 522503, India
[6]School of Engineering Science, Department of Software Engineering, LUT University, 53850 Lappeenranta, Finland

Corresponding author: Mhamad Bakro (mhwb14794@gmail.com)

This work was supported by the King Saud University, Riyadh, Saudi Arabia, through the Researchers Supporting Project under Grant RSP2023R476.

ABSTRACT The focus of cloud computing nowadays has been reshaping the digital epoch, in which clients now face serious concerns about the security and privacy of their data hosted in the cloud, as well as increasingly sophisticated and frequent cyberattacks. Therefore, it has become imperative for both individuals and organizations to implement a robust intrusion detection system (IDS) capable of monitoring packets in the network, distinguishing between benign and malicious behavior, and detecting the type of attacks. IDS based on ML are efficient and precise in spotting network threats. Yet, for large dimensional data sizes, the performance of these systems decreases. Thus, it is critical to building a suitable feature selection approach that selects necessary features without having an impact on the classification process or causing information loss. Furthermore, training ML models on unbalanced datasets show a rising false positive rate (FPR) and a lowering detection rate (DR). In this paper, we present an improved cloud IDS designed by incorporating the synthetic minority over-sampling technique (SMOTE) to address the imbalanced data issue, and for feature selection, we propose to use a hybrid approach that includes three techniques: information gain (IG), chi-square (CS), and particle swarm optimization (PSO). Finally, the random forest (RF) model is utilized for detecting and classifying various types of attacks. The suggested system has been verified by the UNSW-NB15 and Kyoto datasets, achieving accuracies of over 98% and 99% in the multi-class classification scenario, respectively. It was noticed that an intrusion detection system with fewer informative features would operate more effectively. The simulation results significantly outperform other methodologies proposed in the related work in terms of different evaluation metrics.

INDEX TERMS Improved design for cloud-IDS, feature selection, PSO-based metaheuristic, random forest.

I. INTRODUCTION

Nowadays, the progress in digital technologies has led to an explosive growth of cloud computing (CC) [1] applications in different fields due to its services (SaaS, PaaS, and IaaS) and its advantages such as expandability, availability, low cost, and so on [2]. However, this has led to a rising number of threats and created a massive market for cyber security [3]. According to this research [4], companies and organizations faced 50 million cyber assaults in 2010, and by 2019, that figure had increased to 900 million, and the figure is still continuously rising. Both individuals and enterprises have suffered serious damage and big financial losses as a result of these cyberattacks. Based on recent Juniper research [5],

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

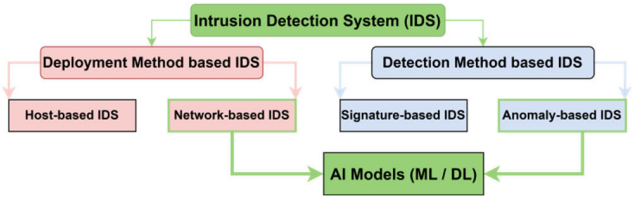

FIGURE 1. IDS classification.

the expense of security breaches is forecast to increase from USD three trillion annually to over USD five trillion in 2024. These immense economic losses made users apprehensive about storing their data in the cloud; thus, the primary goal of the cloud service provider (CSP) is to allay their fears by providing the greatest level of security and maintaining their personal data by investing in cybersecurity solutions. In 2022, the worldwide cybersecurity industry was estimated to be worth USD 202.72 billion. From 2023 to 2030, it is anticipated to increase at a CAGR of 12.3% [6].

The cloud is composed of three primary network types: virtual, internal, and external. Communication among virtual machines (VMs) running on the same physical server/infrastructure is allowed through the virtual network [7]. Various cloud components, such as network servers, management systems, and storage systems, can connect with each other over the internal network. The external network serves as the main communication channel between the cloud user (front end) and the CSP (back end). Altogether, these networks enable the successful delivery of cloud services to customers. Therefore, protecting the network from any potential attack is of utmost importance. The cloud employs a variety of cybersecurity strategies, such as firewalls, intrusion prevention systems (IPS), IDSs, etc., to address numerous security issues. Recently, network threats have worsened due to a lack of adequate counter-security actions [8]. Consequently, IDS is implemented in the cloud model to combat security concerns. With cloud computing services being provided through the internet network, guaranteeing data security and protection is among the greatest obstacles to cloud success. The key security challenge in the cloud is detecting and preventing network intrusions. Given that the network serves as the cloud's backbone, any network vulnerabilities have an immediate impact on the overall security of the cloud [9]. Conventional solutions such as firewalls and even traditional signature-based intrusion detection techniques are no longer effective in confronting intruders [10], as their non-deterministic nature makes them unsuitable for cloud environments. Therefore, it is crucial to develop anomaly-based IDSs using ML models with high accuracy, an elevated DR, and a minimal FAR before implementing this IDS on each server in cloud computing to monitor network traffic for detecting attacks.

IDS is a protective measure that continuously monitors and analyses host and network traffic to determine various types of abnormal activity that breaches security procedures [11].

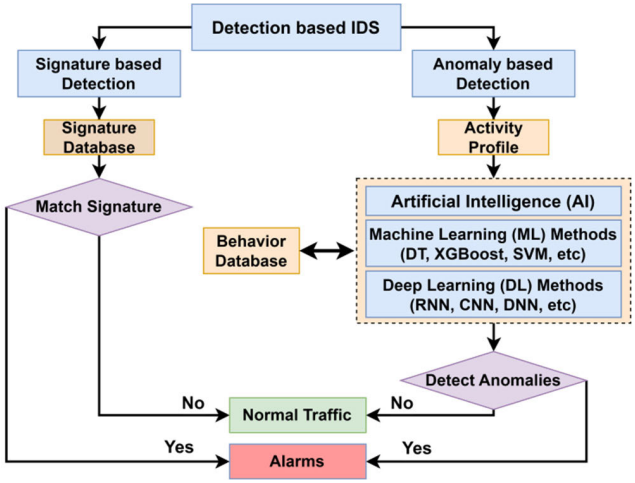

FIGURE 2. Taxonomy of IDS based on the detection method.

IDS can be categorized based on how they are deployed or detected. In Figure 1, the classification is presented.

- The operation-based IDS philosophy is divided into host-IDS (HIDS) and network-IDS (NIDS). HIDS is a single-device security tool that exclusively cares about the security of its host. The key disadvantage is that it must be installed on every host that requires intrusion protection, which adds extra operational costs to every node and lowers the IDSs overall performance. On the other hand, NIDS is established on the network with the purpose of preventing intrusions on all devices included in the network. Since cloud computing services are provided over an Internet network and because NIDS is more comprehensive, we use NIDS datasets for the validation of our proposal.
- Figure 2 shows the detection methods that are categorized into signature-based IDS, also known as misuse detection. The data in the network is matched with the types of attacks in the database of signature-based IDS so that a warning is generated if a match is found, but the main weakness is if in the absence of that there is some attack on the database, the intrusion cannot be identified. On the contrary, normal behavior cases are stored in the database of rule-based systems, also known as anomaly-based IDS, and it monitors all packets on the network to issue a warning if any deviation outside the specified rules occurs. Its main feature is its ability to detect unknown attacks, also known as zero-day attacks [12]. Yet, the key issue is the increased FAR and lower DR since it is challenging to distinguish between benign and malicious profiles for intrusion detection.

Recently, ML-based IDSs have become industry leaders and hold great promise for enhancing the field of intrusion detection studies. ML models offer IDSs the capability of self-learning and improvement from available data. Generally, there are two types of ML models: supervised-ML and unsupervised-ML. In supervised ML, models are trained with

FIGURE 3. Process of building an ML model.

labeled data, but unsupervised ML models use (unlabeled) unstructured data for training. This study uses supervised ML techniques, particularly multi-class classification, to identify various types of attacks. Since datasets are crucial for evaluating IDSs, it is important to utilize enough high-quality and well-pre-processed data. Most often, big datasets with high-dimensional feature sizes are employed to train the ML models, therefore consuming an immense amount of computational resources and causing poor performance of the model. Implementing feature selection techniques is one of the aspects of preprocessing that is used to address the dimensionality issue. Feature selection is the procedure of choosing the optimal subset of relevant features from a high-dimension collection to increase performance, improve classification accuracy, and decrease cost without losing information. There are many approaches to selecting features. Figure 3 illustrates the design of an ML model.

In this paper, we propose a hybrid strategy for feature selection that includes filter methods (IG and CS) and a bio-inspired algorithm (PSO). The combination of these three methods (IG, CS, and PSO) is a novel approach that can provide a more robust feature selection process by exploiting the strengths of each technique to enhance the possibilities of determining the most related features. The unbalanced data negatively impacts the performance metrics, especially with the multi-class classification in the case of the minority classes. It is essential to handle this issue along with maintaining the information values by increasing the minority instances; thus, the SMOTE algorithm is used to do that. The proposed system is evaluated by using a RF as a classifier and a couple of benchmark datasets, namely, UNSW-NB15 and Kyoto. The following constitutes the paper's main contribution:

- Address the imbalance data issue using a SMOTE.
- The proposed methodology, which consists of IG, CS, and PSO, seeks to find an optimal feature subset that not only improves the performance of the model but also contains features that are highly correlated with the target variable and are informative.
- Using a RF as a supervised ML model in the detection of the attacks. Its merits, such as handling both continuous and categorical data, addressing missing and outliers' values, consuming shorter time in training, etc., made it our choice as a classifier.
- Testing the suggested model is conducted using public standard datasets, called UNSW-NB15 and Kyoto, which consider host and network datasets. The experiment focuses on multi-class classification to illustrate the evaluation metrics for each class.

The current study is structured as follows: Section II presents state-of-the-art works related to the IDSs. Brief details about the suggested methodology are provided in Section III. In Section IV, a short explanation is offered related to the used datasets and performance metrics. The simulation process and results discussion are mentioned in Section V. Finally, the conclusion and future works are shown in Section VI.

II. RELATED WORKS

This section provides a brief summary of recent studies that aim to enhance IDS performance through the use of feature selection techniques, including those that are based on bio-inspired or filter-inspired algorithms as well as ML and DL classifiers. Benmessahel et al. [13] created an evolutionary neural network (ENN) using an artificial neural network (ANN) and a novel natural evolutionary algorithm (EA) dubbed the multiverse optimizer (MVO). The major goal of this research was to train a feed-forward multi-layer ANN using an MVO to recognize new threats and resolve the issues that ANNs run into. The MVO-ANN exhibits great efficacy after being verified using the UNSW-NB15 dataset. Yang et al. [14] presented a model comprising a DNN plus an improved conditional variational autoencoder (IC-VAE). IC-VAE has the capability to investigate and learn about possible sparse representations between network data attributes and classes. The learned-VAE encoder is used to adjust the weight of the DNN hidden layer and minimize the size of the input data. The ICVAE decoder is able to balance the training dataset by increasing the records of minority attack types. The DNN functions as both a classifier and a feature extraction model, learning more quickly and easily than conventional MLP networks. ICVAE-DNN was tested by UNSW NB15 and demonstrated high performance, particularly in detecting unknown and minority attacks. Tama et al. [15] designed a new system for IDS based on hybrid feature selection techniques such as PSO, the ant colony algorithm (ACA), and the genetic algorithm (GA), which are used to decrease the dimension space of the training sets. This combination of PSO-ACA-GA determined a subset of 19 features in the UNSW-NB15 dataset. Then, a two-level meta-ensemble classifier, i.e., rotation forest and bagging, achieved considerable performance.

Khan et al. [16] used a stacked auto-encoder (SAE) and a soft-max classifier to develop a unique two-stage deep learning (TSDL) approach for effective network intrusion detection. The proposal includes two decision steps: the first one uses a probability score value to determine if network traffic is benign or malicious. This is then utilized as an added feature for identifying benign states and other types of threats at the final decision step. The suggested model is able to classify data automatically and effectively by learning relevant feature representations from significant volumes of unlabeled data. Their proposal also used the SMOTE algorithm to solve this issue of unbalancing data. Vinayakumar et al. [17] suggested a hybrid intrusion detection alert system that can evaluate host- and network-level

activity utilizing a massively scalable architecture running on commodity computing servers. The system used a distributed deep learning algorithm with DNNs for processing and real-time analysis of extremely large amounts of data. The performance of the DNN model was carefully compared to that of conventional machine learning classifiers using a variety of standard IDS datasets. They found that DNNs surpassed the traditional machine learning classifiers in all situations except for UNSW-NB15, where decision trees (DTs) and random forests performed better overall in multi-class classification. Patil et al. [18] explained a framework for hypervisor-level distributed network security (HLDNS), which is installed on every processor server in a cloud environment. For the purpose of detecting intrusions, a separate server keeps track of the network activity going to and from the virtual, internal, and external networks connected to the underpinning virtual machines (VMs). They were able to recognize both known and unknown threats because they applied both misuse-based and rule-based detection approaches. Furthermore, misuse-based detection was used before rule detection, which reduced the total computing costs because their model just had to scan the network data for zero-day attacks. In order to select feasible attributes from cloud network traffic, they advanced the binary bat algorithm (BBA) with two additional fitness functions. The generated features were then applied to the RF model to detect intrusions, and finally, the detected attacks from various servers were gathered to update the signature (misuse) database.

Saleh et al. [19] proposed an IDS based on a hybrid methodology that combined three techniques, namely, naïve base feature selection (NBFS) for reducing the size of datasets and figuring out the best features, optimized support vector machines (OSVM) to reject outliers, and prioritized k-nearest neighbors (PKNN) in order to identify and classify the threats. The Kyoto dataset was one of three datasets utilized in this approach's testing. By simulation, the proposal proved effective in real-time, and it is ideal for addressing the multi-class classification issue. It can also be used to reduce training and testing times while increasing DR. Zhang et al. [20] suggested an MSCNN-LSTM methodology that consisted of multi-scale convolutional neural networks to analyze the spatial attributes of the dataset and then use a long short-term memory to handle the temporal attributes; thus, the classification was executed using spatial-temporal features. This approach demonstrated its potent capacity for dealing with datasets with high complexity and dimensionality. The UNSW-NB15 dataset, with only 20 features out of 49, was used for verifying the method. Therefore, in the future, they are planning to use feature selection techniques to achieve better performance. Kasongo and Sun [21] proposed using the XGBoost model as a feature selection model to decrease the dimension of the dataset and enhance detection accuracy. Then, they used ML methods (SVM, kNN, LR, ANN, and DT) to classify and detect the threats. Finally, after comparing the performance of ML models, they found that DT and ANN are better in terms of binary and multi-class classification, respectively. In the future, they plan to employ a synthetic oversampling technique to grow the number of minority-type instances to solve the imbalanced data problem.

Kumar et al. [22] provided an integrated classification-based IDS and tested its performance on the offline standard dataset (UNSW-NB15) and online real-time dataset (''RTNITP18'' that have been created by the authors in the CSE lab of NIT Patna). These five classes (normal, probe, DOS, generic, and exploit) have been detected in both datasets. They have used a DT as a classifier and IG model to select the features in the UNSW-NB15 dataset. Almomani [23] designed an intrusion detection system based on bio-inspired feature selection techniques such as firefly optimization (FFA), grey wolf optimizer (GWO), genetic algorithm (GA), and PSO, along with a couple of ML models (J48 and SVM) as classifiers to evaluate his approach using the UNSW-NB15 dataset. The proposal showed heightened results due to employing the feature selection strategy, which impacted the consumed time and improved the accuracy level. His rule-based IDS includes 17 rules for feature selection. In total, 4 rules are formed by every single algorithm (FFA, GWO, GA, and PSO), and 13 sets of rules are formed because of the combination of all of them. Each rule, R, has a different number of features. By the simulation, R17 with 30 features has the most satisfactory effect as it was shown that the performance of J48 was more profitable than SVM, and the PSO results were the best among other individual methods; therefore, we were inspired to use PSO as a feature selection algorithm in the current study. Jiang et al. [24] discussed a unique technique for intrusion detection systems that integrates hybrid sampling plus deep hierarchical networks. Initially, they minimize the noise instances in the majority class using one-side selection (OSS), and then they expand the number of the minority instances using SMOTE. Finally, they created a deep hierarchical network structure by using a CNN to extract spatial attributes and a bi-directional long short-term memory (BiLSTM) to extract temporal attributes. The model produced high-quality results due to using repeated multi-level learning methods and creating a balancing dataset, which allowed the model to comprehensively grasp the attributes of minority instances and drastically reduce the amount of time it takes to train. Rajesh Kanna and Santhi [25] suggested an OCNN–HMLSTM model which combined the optimized CNN (OCNN) for picking up the spatial features that employed the lion swarm optimization (LSO) to adjust the hyperparameters to obtain a perfect setting, along with a hierarchical multi-scale LSTM (HMLSTM), which has superior effectiveness for temporal attributes; in addition, it also classifies packets of the network. This approach's ability to automatically comprehend spatial-temporal characteristics makes it efficient for detecting threats. However, the complexity of the model due to using deep learning methods led to a long training time (30,665 s), which meant the consumption of a lot of resources, which is what we

have examined in this study. The authors intend to investigate feature selection procedures in the future.

Sreelatha et al. [26] presented an efficient cloud IDS thanks to the feature selection concept and classification. Based on the sandpiper optimization algorithm (SOA), the relevant and valuable attributes are selected from the provided incursion dataset with the least amount of information lost. The extended equilibrium deep transfer learning (EEDTL) method is then used to categorize various threats according to the best features that were chosen by SOA. Transfer learning employs a pre-trained network called AlexNet, which is considered the most common structure in DCNN, to well-configure the characteristics in the convolution layers. To update the network weights, the extended equilibrium optimizer (EEO) is also employed. The simulation outcomes proved that the proposal demonstrated superior performance compared to other methodologies, but still, there are better results in terms of DR and precision. In upcoming works and to enhance the effectiveness, they plan to carry out feature selection based on hybrid optimization algorithms along with hybrid machine learning strategies. Kanna and Santhi [27] built an efficacious IDS based on the hybrid-optimized deep learning involvement of ABC-BWO-CONV-LSTM. The artificial bee colony (ABC) method selects features as the initial step. The next step is the classification step using the hybrid DL model of black widow optimized convolutional long short-term memory (BWO-CONV-LSTM) established on a MapReduce framework. The proposal consists of CNN and LSTM to understand the spatial and temporal attributes, and BWO to tune the hyper-parameters optimally. The experimental outcomes showed that the suggested strategy performed remarkably well in intrusion detection with the least amount of spatial loss and architectural complexity; it also handled the issues of overfitting and class imbalance. However, the time of training (26,721.2 s) and testing (402.67 s) is still long, which implies significant resource usage.

By investigating the previous literature and looking at the studies in Tables 8 and 9, we have found some limitations, such as not handling the issue of data imbalance, using a small number of data samples, and using the Kyoto dataset with only two cases (normal traffic and known attack) in the evaluation phase. Additionally, some strategies have employed deep learning models that can produce good results but require significant computational resources. Moreover, some of these approaches are complex and challenging to interpret, making it difficult to understand their inner workings and how they improve their results. Furthermore, we have realized that to obtain an intrusion detection system with satisfactory effectiveness, the dataset must be well prepared. Therefore, we summarize our improvement steps as follows. Initially, treat the imbalance issue by using one of the methods, such as SMOTE, due to its good impact on the results, especially in the case of minority classes. Then, using feature selection techniques for reducing the size of the dataset by determining the correlated and valuable features and eliminating the meaningless attributes without losing any information. We have observed that the methods of feature selection were based on wrapper, filter, or bio-inspired strategies, which played a dual significant role in either selecting the best collection of attributes or enhancing the weights of neural network classifiers. Finally, we observed that although deep learning models demonstrate great performance when employed for classification, the consumption of computing resources is still huge. Although a lot of related research utilized tree structure models as classifiers because of their numerous merits, ease of use, reasonable cost, and simplicity.

However, we wish to draw your attention to the fact that we selected relevant works that used the same datasets as ours for comparison, although there may be other studies that used different approaches, but with other datasets.

III. METHODOLOGY

Before deploying an IDS in the cloud, we must preprocess the datasets used to train it. These datasets are quite large and encompass a wide range of attacks, along with a significant amount of unrelated information. Therefore, we must select the most relevant features, which will later be used to train the classifier. This classifier will then differentiate between benign packets and diverse types of attacks. So, in this section, we will provide brief theoretical details of our vision. We sought to employ the strengths of previous related works; thus, we prepared the datasets well. The SMOTE algorithm was used, combining filter-based (IG and CS) along with bio-inspired-based (PSO) procedures in feature selection. Finally, the classification stage was conducted using RF model. The working stages of our suggested model can be seen in Figure 4.

A. DATA PREPROCESSING

The datasets include a wide range of categorical and numerical data, etc. Some of these data may have skewed and irregular values, which provide undesirable outcomes. This section pre-processes the data in order to prepare it for feature selection because ML algorithms only take numeric and cleaned feeds. Initially, the worthless features and null values were removed, along with any unnecessary data [28]. Then, we took the following actions:

- Feature encoding refers to the conversion of categorical data into a numerical input. The majority of the attributes in this study are numerical, whereas the remainder are categorical. Thus, the categorical attributes must be transformed to fit the ML model for training. Label encoding and one hot encoding are the methods that are most frequently employed for feature mapping, and these techniques have their own advantages and disadvantages. The second strategy offers better performance but greatly increases the number of attribute dimensions [29]; hence, we have chosen to employ the first method where we find good results with it [30]. Combining label encoding with one-hot encoding definitely

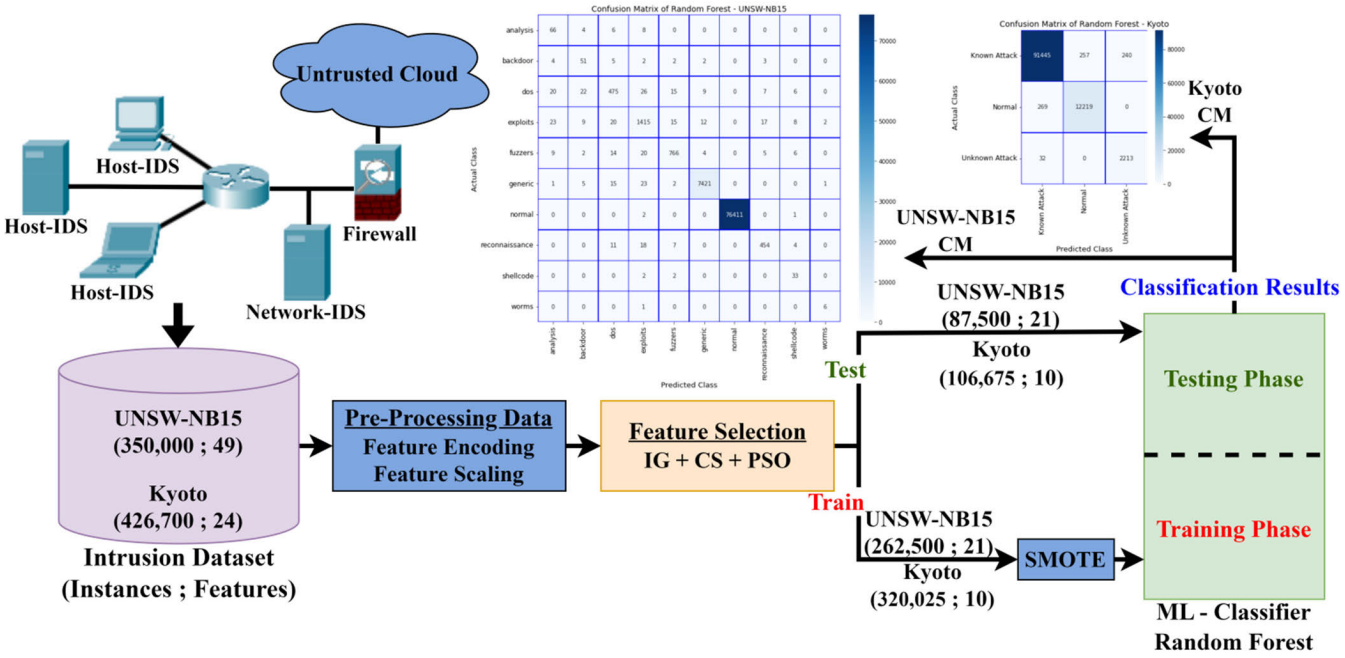

FIGURE 4. Overview of the suggested intrusion detection system based on a hybrid of feature selection methods.

increases the number of features and makes the dataset larger, which leads to increased resource consumption. Therefore, it is better to stick with label encoding alone, as our goal is to reduce the number of features. Here is a list of categorical features that have been encoded using the label encoding technique: For the UNSW-NB15 dataset, they are ('srcip', 'dstip', 'proto', 'state', and 'service'), and for the Kyoto dataset, they are ('Service', 'Flag', 'IDS_detection', 'Malware_detection','Ashula_detection','Source_IP _Address','Destination_IP_Address', 'Start_Time', 'Protocol'). As can be observed, there are 5 and 9 categorical features in the UNSW-NB15 and Kyoto datasets, respectively.

- Feature scaling: this is a method for normalizing and transforming all feature values into a predefined range. It is an essential step since it eliminates the biased attributes of higher values. The popular approaches to feature scaling are standardization, which may be called the Z-score, and normalization, which is also known as min-max scaling, which often offers gratifying results [23]. We employed the min-max approach, which is given as follows:

$$H = \frac{H - H_{min}}{H_{max} - H_{min}} \quad (1)$$

where H_{min} and H_{max} show the minimum and maximum values of feature H.

The optimal features will be selected from the pre-processed dataset in step B. Following that, the training dataset and the testing dataset will be produced from the pre-processed data. The data imbalance issue impairs the effectiveness of the classifiers, specifically in the minority categories of threats; therefore, we used SMOTE to cope with this problem, where the training dataset's minority class records are increased [24]. Finally, the classifier will be trained using the new balanced training dataset and will be tested by the testing dataset, as we will see in step C.

B. FEATURE SELECTION

The performance of the proposal can be enhanced by using feature selection as a pre-treatment step. A reduced dataset offers shorter training times and more accurate efficiency. We chose the filter-based feature selection method (IG and CS) due to its many advantages, including its interpretability, flexibility, scalability, and time complexity. Meta-heuristic techniques such as PSO combined with filter methods produce a more robust optimum feature subset. In light of this, we will concisely discuss our feature selection strategies.

1) INFORMATION GAIN IG

IG is a measure of the decrease in entropy after a feature is selected. IG can be used to determine which features are most important for making predictions about the target variable. Features with a high IG are considered to be more informative than features with a low IG. This measure allows us to rank the features and choose the most important ones for the final effective feature set; hence, the features with a high IG were kept in the model, while features with a low IG were removed [31]. The IG of a feature can be calculated using the

formula [22]:

$$IG\,(B) = E\,(A) - E_B\,(A) \tag{2}$$

where A denotes a dataset's size and B is a feature.

2) CHI SQUARE CS

CS is a statistical method that determines the correlation between 2 variables. The CS test is used in feature selection to identify how much each feature depends on the target variable. Features with a high CS score demonstrate a strong reliance on the target variable and are deemed significant features [32]. The formula of the CS test is shown by [31]:

$$\mathrm{CS} = \sum\nolimits_{\mathrm{xz}} (\mathrm{I_{xz}} - \mathrm{J_{xz}})^2 \Big/ \mathrm{J_{xz}} \tag{3}$$

where x and z are 2 variables and I and J, respectively, stand for the observed value and expected value.

3) PRACTICAL SWARM OPTIMIZATION PSO

It belongs to the bio-inspired artificial swarms' intelligence (SI) families, which replicate the naturally intelligent behavior of animals or insects to address a naturalistic problem [33]. PSO is a global optimization technique that imitates the foraging behavior of birds, and it is used to solve many real-world issues [34]. PSO is the most widely used among the SI algorithms because it delivers the best solutions in a reasonable amount of time [35]. In this approach, the feature set is birds, which are represented by particles scattering across hyperspace that continuously search for the best global locations (*Gbest*). Finding the *Gbest* is instructed by the local particle's best position (*Pbest*). A *Pbest* is tuned if the particle discovers another best location [23]. Therefore, the optimized value (the global best value) is produced as a result of each particle's cooperation mechanism.

Each particle has a random location (L_i) and speed (S_i). Assume that z_1, z_2, and w are constants that stand for cognitive learning, social learning, and inertia weight, respectively. Additionally, $Pbest_i$ is the personal best position of the particle i, and *Gbest* is the global position among the particles. Suppose n_1 and n_2 be random values. Thus, the key rules for adjusting each particle's location and speed are as follows [15]:

$$L_i\,(t+1) = L_i\,(t) + S_i\,(t+1) \tag{4}$$

$$S_i\,(t+1) = wS_i\,(t) + z_1 n_1\,(Pbest_i - L_i\,(t)) + z_2 n_2\,(Gbest_i - L_i\,(t)) \tag{5}$$

4) PSEUDOCODE

The following is the pseudocode for a feature selection method that combines IG, CS, and PSO:

Algorithm 1 Feature Selection

1. Initialize the population of particles with the features set.
2. Calculate the information gain and chi-square values for each feature in the dataset.
3. For each particle in the population:
   a) Evaluate the fitness of the particle using the information gain and chi-square values.
   b) Update the speed and location of the particle based on the particle swarm optimization algorithm.
4. Select the best-performing particle from the population.
5. Repeat steps 3 and 4 for a predefined number of iterations or until a stopping criterion is met.
6. Return the best feature subset as the final result.

5) DIFFERENCES BETWEEN THE SELECTED METHODS

Despite the apparent overlap between IG and CS methods in terms of filtering out features based on their relevance to the target variable, it's important to recognize that they each have strengths that the other lacks. There are subtle differences between these methods, which can be summarized as follows:

The two methods are grounded in distinct statistical theories and function differently. The IG method measures the reduction in entropy, which refers to the decrease in uncertainty or randomness. This reduction is achieved by partitioning the samples based on a specific feature. In turn, this provides a measure of the effectiveness of the selected feature in data classification. IG excels at identifying and selecting features that help reduce uncertainty across a wide range of feature types. Conversely, the CS test, a non-parametric method, assesses the presence of a significant association between two categorical variables. This method operates under the assumption of the independence of these variables. While the CS method excels at identifying relationships between categorical variables, it may not always be the most suitable for continuous or discrete features.

The integration of these methods with PSO forms a crucial part of our strategy to effectively harness the strengths of each individual method while counteracting their inherent weaknesses. As demonstrated in Table 6, the fusion of these methods, rather than exclusively depending on a single one, leads to the creation of a superior set of features. This results in significant improvements in the overall performance of our intrusion detection system.

C. CLASSIFICATION

The purpose of classifiers is to categorize the received packets as benign or malicious, and for performing this function, ML and DL models are used. DL models are more complicated and require more resources; thus, it is better to use ML methods in our case, as they have enough power to handle our datasets. ML classifiers have many models such as SVM, RF, etc., but by experiment, we found that the performance of RF is good and takes less time in training and testing compared with SVM and other models. Therefore, the RF classifier was employed in the current study, which was fed with the best optimal feature set.

◇ **Random Forest RF**

RF is an ensemble learning approach that is employed to predict outcomes based on DT. DT is a supervised ML model employed for regression and classification. A tree-like struc-

ture is used to present the potential outcomes or decisions, in which each node stands for an attribute or feature, every branch denotes a probable value of the feature, and every leaf represents a class label or a numerical value. The tree is constructed using a training set [11]. The objective of this model is to build a tree that properly predicts the target variable while maximizing the information gain and minimizing the number of nodes [22]. Thus, to prevent overfitting, the DT model automatically chooses the most beneficial attributes for constructing a tree, and it also performs a pruning process to eliminate unnecessary branches. The most popular types of DT are ID3, C4.5, and CART [36]. Sometimes, one tree may not be sufficient to yield high performance; therefore, XGBoost and RF, which form from multiple DT, are used [30]. A RF generates multiple decisions, making it suitable for handling extremely large datasets; it provides an estimation of the variables that are most significant in the classification process. In this research, we observed improved performance when using the RF method. As this approach combines multiple DT, it results in a more accurate and robust model.

D. POTENTIAL LIMITATIONS OF THE METHODOLOGY

The proposed methodology involves four main steps: preprocessing, feature selection, balancing the training dataset, and applying a random forest classifier. Each of these steps has its own limitations and potential sources of error.

- Preprocessing: the first step is to preprocess the data by encoding categorical variables and scaling numerical data. One limitation of this step is that label encoding may not be appropriate for all categorical variables. In some cases, one-hot encoding may be a better approach. Additionally, min-max scaling can be sensitive to outliers, which can affect the performance of the classifier.
- Feature selection: the second step is to select the relevant features from the pre-processed data using IG, CS, and PSO. One limitation of this step is that it may result in overfitting if the feature selection is not performed correctly. Additionally, some important features may be overlooked, leading to underfitting.
- Balancing the training dataset: It's worth noting that the issue of imbalanced datasets is prevalent in many real-world machine-learning applications, especially in the field of network intrusion detection. The third step in our proposal is to balance the training dataset using SMOTE. While SMOTE is an effective technique for balancing datasets, it may also introduce noise into the dataset due to the significant oversampling of minority classes, leading to overfitting. We completely agree with this potential, but this is a known trade-off when addressing class imbalance through oversampling. However, we have adopted strategies to mitigate these potential drawbacks. We used a hybrid feature selection method, combining IG, CS, and PSO, which has effectively allowed us to filter out the noise and retain only the most informative features. Furthermore, we used a RF classifier, which is known for its resistance to noise and overfitting. Concerning the testing dataset, we do not balance it in the same way as the training dataset. Our approach is grounded on a commonly accepted principle in machine learning: the testing dataset should as closely as possible reflect the real-world data distribution to provide an accurate indication of the model's performance in a practical and realistic environment. Consequently, we have maintained the same class proportions in our testing dataset as those found in the original used dataset. Additionally, SMOTE may not be suitable for all datasets, and other balancing techniques may be more appropriate, such as random oversampling or adaptive synthetic oversampling (ADASYN), which is considered an advanced version of SMOTE. In future studies, we plan to handle class imbalance by employing other techniques belonging to these strategies: over-sampling the minority class, under-sampling the majority class, and using a combination of both. We will then evaluate the effect of these techniques on performance.
- Random forest classifier: the fourth step is to apply a RF classifier on the balanced preprocessed dataset. One limitation of RF is that it can overfit if the number of trees is too high. Additionally, RF may not perform well on datasets with high dimensionality. Thus, the performance of the RF classifier is sensitive to the choice of hyperparameters, such as the number of trees and so on.

Generally, every methodology, regardless of its steps, has possible restrictions and sources of error that may be related to the quantity, quality of data used, or any other factors. However, experience remains the best evidence to confirm or deny the possibility of error. Based on the simulation based on the data used, our approach has proven to be effective compared to existing conventional studies.

IV. DATASETS AND EVALUATION METRICS

An overview of the employed datasets and evaluation criteria is provided in the following subsections.

A. DATASETS DESCRIPTION

Producing a comprehensive dataset is a costly process that requires significant financial resources and specialized knowledge. Consequently, one of the significant challenges faced by the IDSs was the systematic generation of an up-to-date dataset that encompasses a wide range of threat types and reflects the actual environment. UNSW-NB15 and Kyoto are considered labeled network traffic datasets and are widely used for evaluating both the NIDS and HIDS. Table 1 presents the statistics of the intrusion datasets used in this study.

To support the research society, the dataset should be frequently revised to include the latest common attack instances.

TABLE 1. Statistical of used intrusion datasets.

Dataset Name	Total No. Of Features	Total No. Of Records	Dataset Source
UNSW-NB15	49	350,000	[37]
Kyoto	24	426,700	[38]

TABLE 2. Partial training and testing sets of UNSW-NB15.

No	Category Name	Used Dataset	Training Dataset	New Training Dataset	Testing Dataset
0	Normal	305,554	229,140	229,140	76,414
1	Generic	29,741	22,273	229,140	7468
2	Exploits	6263	4742	229,140	1521
3	Fuzzers	3287	2461	229,140	826
4	DOS	2272	1692	229,140	580
5	Reconnaissance	1966	1472	229,140	494
6	Analysis	393	309	229,140	84
7	Backdoor	304	235	229,140	69
8	Shellcode	197	160	229,140	37
9	Worms	23	16	229,140	7
Total		**350,000**	**262,500**	**2,291,400**	**87,500**

While the aforementioned two datasets used in the study reflect crucial attack types, it is acknowledged that they may not represent all possible network traffic scenarios. A brief recap of each one of them is provided as follows:

1) UNSW-NB15

This dataset was produced by Moustafa et al. [39] in the Australian center for cyber security at the University of New South Wales (UNSW) to address the problems seen in the NSLKDD and KDDCup 99 datasets [17]. It took 31 hours and a variety of tools to gather 100 GB of data, which includes around 2.5 million instances [23]. The total number of features is 49, which were collected via Argus tools, Bro-IDS, and 12 newly developed algorithms. Table 15 displays each feature with its corresponding data type. It reveals that out of the total, 43 features are numerical, while 6 are nominal. These features were categorized into five groups: basic features, flow features, time features, content features, and additional features. Two features function as tags: ''attack_cat'', which can take these values ('generic', 'exploits', 'fuzzers', 'dos', 'reconnaissance', 'analysis', 'backdoor', 'shellcode', 'worms', and 'normal'), and ''label'', which takes 1 for an attack and 0 for normal traffic. There are nine types of attacks. Our target variable in this study is the ''attack_cat'' feature. As usual, just a portion of the dataset is used owing to its large size [36]. Table 2 illustrates the number of used records in the sets of training and testing for each category, as well as the new training set generated via SMOTE, which increased the number of training instances to correspond with the number of records in the majority class, which in this case is 229,140.

TABLE 3. Sets used from Kyoto for training and testing.

No	Category Name	Used Dataset	Training Dataset	New Training Dataset	Testing Dataset
1	Normal	49,794	37,306	275,756	12,488
-1	Known attack	367,698	275,756	275,756	91,942
-2	Unknown attack	9208	6963	275,756	2245
Total		**426,700**	**320,025**	**827,268**	**106,675**

TABLE 4. Confusion matrix.

		Predicted	
		Attack	Normal
Actual	Attack	TP	FN
	Normal	FP	TN

2) KYOTO

This dataset was created in real-time by Song et al. [40] at Kyoto university in Japan between 2006 and 2015. In total, 19,683 MB of network traffic were collected from darknet sensors, honeypot, web crawler, email servers, and other servers. The dataset includes 24 statistical attributes, 14 of which are taken from the KDD Cup'99 dataset, while the remaining 10 are modern attributes [17]. Table 16 provides a representation of each feature along with its associated data type. From the total number of features, it can be discerned that 15 features are numerical and 9 are of a categorical nature. The instances of traffic are categorized into three types based on a feature called ''Label'', which has these values 1, −1, −2, which refer to a normal packet, a known attack, and an unknown attack, respectively. This ''Label'' feature serves as our target variable. Table 3 shows the number of utilized instances in the sets of training and testing for each type, along with the new training set produced by employing SMOTE, which has led to a rise in the number of training samples to equal the number of records in the majority class, which in this case is 275,756.

B. PERFORMANCE METRICS

We evaluate the performance of our suggested model against the conventional studies in terms of accuracy, recall, precision, F-measure, and FAR, particularly since these metrics are commonly utilized for assessing intrusion detection models and are calculated using the confusion matrix (CM) as shown in Table 4 [11]:

- True Positive TP: an intrusion instance is properly determined as an attack.
- True Negative TN: a normal instance is properly determined as normal traffic.
- False Positive FP: a normal instance is wrongly determined as an intrusion.
- False Negative FN: an intrusion instance is wrongly determined as normal traffic.

TABLE 5. Models' parameters value.

Model	Parameter value	Model	Parameter value
RF	n_estimators=30 random_state=42	PSO	n_particles=25 n_iterations=50
XGBoost	n_estimators=30 random_state=42	SVM	kernel='linear' gamma='auto'
DT	random_state=42	LR	solver='lbfgs'

Accuracy: This measure refers to the ratio of instances that have been identified correctly out of all instances.

$$\text{Accuracy(ACC)} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{6}$$

Precision: This metric assesses the ratio of attacks that were predicted correctly in relation to the total number of instances that were attacked.

$$\text{Precision(P)} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{7}$$

Recall: This term describes the ratio of instances that were correctly identified as attacks to the total number of instances that were actually attacked.

$$\begin{aligned}\text{Recall(R)} &= \text{Detection Rate (DR)} = \text{Sensitivity (S)} \\ &= \text{True Positive Rate (TPR)} = \frac{\text{TP}}{\text{TP} + \text{FN}}\end{aligned} \tag{8}$$

F-measure: This measurement for assessing the proficiency of a system by considering both its recall and precision.

$$\text{F1} - \text{score} = \text{F} - \text{measure(F)} = \frac{2}{1/\text{Precision} + 1/\text{Recall}} \tag{9}$$

False Alarm Rate: This denotes the percentage of attack instances that were wrongly predicted among all actual normal instances.

$$\text{FAR} = \text{False Positive Rate (FPR)} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{10}$$

Hence, the ACC, DR, and FAR are the significant measures that distinguish any IDS and determine its power. Additionally, we take into account the receiver operating characteristic (ROC) curve, which is produced by contrasting the FPR and the TPR of the model [28]. The ROC curve concept is often used in evaluating the performance of binary classification models [17], but we will use it in multi-class classification.

V. EXPERIMENT AND RESULTS DISCUSSION

In this section, the results of our proposal are presented, discussed, and evaluated against the findings from prior studies. The experiments were run on an Intel Corei5 processor using Python and on the Google Co-laboratory Pro platform, which includes 25 GB of RAM. Sklearn and many other libraries are used throughout the execution.

Table 5 outlines the parameters used in our current work. For our present study, we have achieved satisfactory results without the need for hyperparameter tuning. However, we aim to further enhance these models in the future by employing hyperparameter optimization via the grid search technique. This technique exhaustively tests every possible combination of parameter values to identify the optimal set of hyperparameters, unlike the random search method, which selects values arbitrarily, thus not necessarily yielding the best results. By adopting a grid search strategy, we anticipate an improvement in our model's performance.

The evaluation criteria for our suggested multi-class classification are derived from simulations across various feature selection methods, with all cases utilizing the RF model as a classifier. These values are demonstrated in Table 6, which provides the proposal's results, including its accuracy, the weighted average of precision, recall, and F1-measure, in addition to the average of FAR.

Feature selection was carried out with both filter and bio-inspired techniques, which provided us with a highly informative and related feature set. Initially, the information gain method selected the informative features; then, the chi-square test determined the features that were highly correlated with the target variable; and finally, the PSO algorithm optimized the feature selection process by identifying the best combination of features. The optimal feature set was fed to the random forest classifier, which provided remarkable performance with a reasonable consumption of time and computational resources. It is essential to mention that the efficiency of the proposed method in a real-world setting is unknown due to the evaluation being conducted in lab circumstances according to used datasets, where various factors could affect the model's performance in real-world conditions.

Table 6 presents the performance comparison of our hybrid approach against each standalone feature selection method, all utilizing the RF classifier. The results indicate that our approach consistently outperforms the standalone methods in terms of classification accuracy, precision, recall, F1 score, and FAR. The 'Selected Features' column lists the digits of the features chosen by each approach. To identify the actual name of each selected feature, you can refer to Tables 15 and 16.

The motivation behind developing a hybrid feature selection approach lies in the nature of high-dimensional data and the inherent limitations of each standalone feature selection method. A high-dimensional dataset often includes complex patterns that may not be adequately detected by a single feature selection method. Each method possesses its own strengths and weaknesses. Therefore, our hybrid approach is specifically designed to amalgamate the strengths of these three methods: IG, CS, and PSO. By doing so, it effectively overcomes their individual limitations. IG and CS are statistical methods that can efficiently detect useful patterns in the data, while PSO is an evolutionary algorithm that can explore the feature space in a more global and robust way, thereby capturing complex relationships that might be missed by the statistical methods. We believe that the adoption of

TABLE 6. Outcomes of the proposal.

Dataset	Feature Selection	Selected Features	Total No. of Features	ACC	P	R	F	FAR
UNSW-NB15	Without	All features except No 48	48	96.49	96.58	96.49	96.49	0.161
	IG	1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 15, 16, 18, 23, 24, 32, 37, 49	18	98.02	98.11	98.02	98.08	0.129
	CS	1, 2, 3, 4, 6, 10, 11, 14, 15, 16, 19, 20, 21, 22, 23, 24, 27, 29, 30, 33, 34, 35, 37, 39, 41, 42, 43, 44, 45, 46, 47, 49	32	97.61	97.76	97.61	97.73	0.156
	PSO	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 15, 16, 18, 23, 24, 29, 30, 31, 32, 37, 45, 46, 47, 49	25	97.99	98.07	97.99	98.05	0.141
	IG-CS-PSO	1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 15, 16, 18, 23, 24, 29, 30, 32, 37, 46, 49	**21**	**98.39**	**98.54**	**98.39**	**98.46**	**0.046**
Kyoto	Without	All features except No 18	23	97.32	96.87	97.32	96.50	0.122
	IG	1,2,3,4,5,6,8,10,13,14,17,19,20,21,22,23	16	98.96	98.99	98.96	98.95	0.053
	CS	6, 8, 10, 12, 13, 14, 17, 21, 24	9	98.97	99.01	98.97	98.99	0.039
	PSO	1, 2, 3, 4, 8, 10, 14, 17, 19, 20, 21, 22, 24	13	99.08	99.13	99.08	99.11	0.012
	IG-CS-PSO	**1, 2, 3, 4, 10, 14, 19, 20, 21, 22**	**10**	**99.25**	**99.27**	**99.25**	**99.26**	**0.008**

TABLE 7. Comparison between our proposal using RF and Other ML Models.

Modle	ACC	R	FAR	Trianing Time	Testing Time
UNSW-NB15 / IG-CS-PSO / 21 Features					
LR	97.11	97.11	0.291	124.741 s	0.029 s
SVM	97.83	97.83	0.221	16,823.152 s	1549.362 s
DT	98.12	98.12	0.167	88.021 s	0.014 s
XGBoost	98.38	98.38	0.162	2360.165 s	0.530 s
RF	**98.39**	**98.39**	**0.046**	**328.511 s**	**0.361 s**
Kyoto / IG-CS-PSO / 10 Features					
LR	93.82	93.82	9.101	15.758 s	0.006 s
SVM	95.96	95.96	5.229	11,609.284 s	823.682 s
DT	98.97	98.97	0.082	8.777 s	0.024 s
XGBoost	99.19	99.19	0.061	134.846 s	0.498 s
RF	**99.25**	**99.25**	**0.008**	**64.273 s**	**0.386 s**

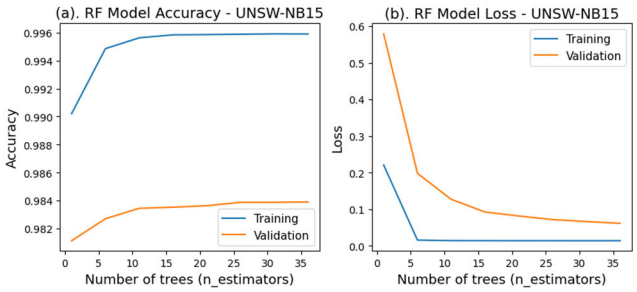

FIGURE 5. Model accuracy and loss - UNSW-NB15.

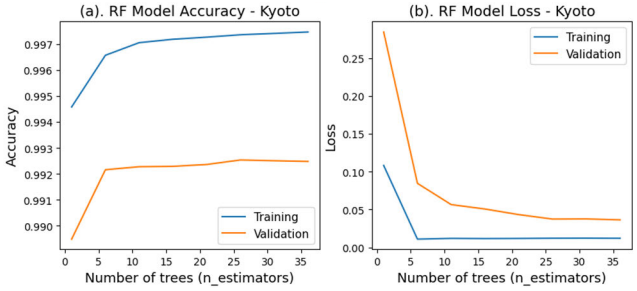

FIGURE 6. Model accuracy and loss - Kyoto.

this hybrid approach is justified as it provides an optimal subset of the original features, thus enhancing the overall performance of the system. Furthermore, its superiority over existing approaches is evident in Tables 8 and 9, demonstrating the robustness of our hybrid method.

Table 7 compares the performance of several machine learning algorithms, namely LR, SVM, DT, XGBoost, and RF. This comparison is based on key evaluation metrics, including ACC, DR, and FAR, as well as both training and testing time. Please note that the specific parameters used by these classifiers are detailed in Table 5. The timestamps marking the start and end of the training and testing phases were determined using the DateTime library.

Table 7 highlights the superior performance of the RF model in comparison to other machine learning models. Our analysis indicates that tree models, namely RF, XGBoost, and DT, excel in terms of ACC, DR, and FAR, as well as training and testing time. Nevertheless, it's worth noting that the SVM model demanded significantly more processing time compared to the others.

Figures 5 and 6 illustrate the training and validation accuracy, as well as the training and validation loss, of our proposal. These values are contingent on the number of trees that comprise the RF model. This contrasts with the neural network models, where, to the best of our knowledge, values are typically determined based on the number of epochs.

TABLE 8. Comparative of the suggested approach with state-of-the-art techniques that use the UNSW-NB15 dataset.

No	Ref. No. and Year	Feature Selection	Methods	No. Of Features	ACC	P	R	F	FAR
1	[41] 2016	-	DT	-	85.56	-	-	-	15.78
2	[42] 2017	PCA	GMM	10	96.70	-	95.60	-	3.5
3	[43] 2017	-	Ramp-KSVCR	-	93.52	-	98.68	98.72	2.46
4	[44] 2017	Weka-ML	RF	5	82.99	81.20	83.00	81.4	0.061
5	[36] 2017	GA-LR	DT	20	81.42	-	-	-	6.39
6	[45] 2017	PCA	GAA-ADS	15	92.8	-	91.30	-	5.1
7	[46] 2017	RFE-CS	RF	16	95.09	-	-	-	2.415
8	[47] 2018	IG	2-stage classifier	-	85.78	-	-	-	15.64
9	[48] 2018	CFS	ABC-AFS	6	95.00	-	88.00	-	2.1
10	[49] 2018	DBN (FE)	Ensemble SVM	-	-	90.47	97.21	93.72	-
11	[50] 2018	-	Parallel K-medoids + KNN	-	94.00	93.4	91.6	-	6.5
12	[51] 2018	FI	RF	11	75.66	75.00	76.00	73.0	-
13	[13] 2018	-	PSO-ANN	-	91.87	-	98.61	-	0.0186
14	[14] 2019	-	ICVAE-DNN	-	89.08	86.05	95.68	90.61	19.01
15	[15] 2019	PSO-ACA-GA	2-stage ensemble	19	91.27	91.60	91.30	-	8.90
16	[16] 2019	TSDL(D-SAE)	Softmax	10	89.13	-	-	-	0.7495
17	[17] 2019	-	2-layers DNN	-	66.00	62.30	66.00	59.60	-
18	[18] 2019	BBA (FSFF+CAFF)	RF	26	97.09	-	95.53	-	2.03
19	[52] 2019	A-PCA	I-ELM	-	70.51		77.36		35.09
20	[20] 2020	-	MSCNN-LSTM	20	95.60	-	-	-	9.8
21	[21] 2020	XGBoost	ANN	19	77.51	79.50	77.53	77.28	-
22	[22] 2020	IG	DT	13	84.83	-	-	-	2.01
23	[23] 2020	PSO-FO-GO-GA	J48	30	90.48	84.14	97.14	90.17	14.95
24	[24] 2020	-	OSS-SMOTE + CNN-BiLSTM	-	77.16	82.63	79.91	81.25	-
25	[53] 2020	FE	DBN	-	85.73	-	-	-	-
26	[54] 2020	NSGA2-MLR	NBTree	11	66.00	-	64.90	-	3.85
27	[55] 2020	CFS	ANN	33	96.44	-	50.40	-	-
28	[56] 2020	WFEU	FFDNN	22	77.16	-	-	-	-
29	[25] 2021	-	OCNN(LSO)–HMLSTM	-	96.33	100	95.87	98.13	5.87
30	[57] 2021	-	MFFSEM	-	88.85	93.88	80.44	86.64	2.27
31	[58] 2021	TS	RF	16	83.12	-	-	-	3.7
32	[26] 2022	SOA	EEDTL	-	99.91	94.93	96.06	-	0.008
33	[27] 2022	ABC	BWO-CONV-LSTM	36	98.67	100	98.78	98.77	4.48
34	[30] 2022	Selectkbest	Stacking (DT-RF-XGBoost)	20	94.00	-	94.00	-	0.06
Our Proposal		**IG-CS-PSO**	**RF**	**21**	**98.39**	**98.54**	**98.39**	**98.46**	**0.046**

As Figure 5 represents, the model reaches its peak accuracy of 98.39% and its lowest loss of 0.066% at the 30th tree. Figure 5a provides a more detailed view: the training accuracy curve starts at 99.05%, gradually rises to 99.59%, and then stabilizes. Meanwhile, validation accuracy begins at 98.15% and peaks at 98.39% with the 30th tree. Accordingly, the difference between the peaks of training and validation accuracy is 1.2.

As depicted in Figure 5b, the training loss starts at 0.24% and gradually decreases to 0.013%, at which point it stabilizes. Conversely, the validation loss commences at 0.59% and declines until it reaches a minimum of 0.066% at the 30th tree, where it too stabilizes. Consequently, the difference between the minimum values of training and validation loss is 0.053.

Similarly, Figure 6 shows the model reaching its peak accuracy of 99.25% and its minimum loss of 0.037%, both at the 30th tree. Figure 6a presents an overview of the accuracy metrics: the training accuracy begins at 99.46%, incrementally rises to 99.74%, and then stabilizes. In parallel, the validation accuracy starts at 98.95% and reaches its peak of 99.25% at the 30th tree. As a result, the disparity between the peaks of training and validation accuracy is 0.49. In contrast, Figure 6b showcases the pattern of the training loss convergence, initiating at 0.21% and gradually descending to 0.012%, at which point it stabilizes. On the other hand, the

validation loss begins at 0.259% and decreases until it reaches its lowest point of 0.037% at the 30th tree, at which point it also stabilizes. Therefore, the gap between the minimum values of the training and validation loss amounts to 0.025.

Overall, as seen in Figures 5 and 6, and Table 7, Tree No. 30 delivered satisfactory results in terms of both accuracy and loss rates, as well as in training and testing times. While we could increase the number of trees, the corresponding improvements in accuracy and loss rates would be marginal and might not justify the additional computational time required.

It's important to note that in the case of underfitting, both the training and validation accuracies are low, while both losses are high. This suggests that the model hasn't learned the training data effectively. Conversely, in the case of overfitting, the training accuracy is very high and the training loss is very low, but the validation accuracy is much lower and the validation loss is much higher. This indicates that the model has learned the training data too thoroughly, resulting in poor performance when applied to new data. In a well-trained model capable of making accurate predictions on unseen data, the training and validation plots should follow similar trends, displaying high accuracy and low loss for both training and testing sets. Nevertheless, there may be a gap between the two plots, reflecting the discrepancy in performance between the training and testing data.

Based on Figures 5 and 6, we have found that the gap between the peaks of training and validation accuracy is 1.2 for the UNSW-NB15 dataset and 0.49 for the Kyoto dataset. Moreover, the difference between the minimum values of training and validation loss is 0.053 for the UNSW-NB15 dataset and 0.025 for the Kyoto dataset. Therefore, our model is well-trained and can make accurate predictions on unseen data. It performs better on the Kyoto dataset than on the UNSW-NB15 dataset.

However, as can be inferred from Figures 5 and 6, and in light of the aforementioned details, our model demonstrates signs of very slight overfitting, particularly with the UNSW-NB15 dataset. This issue needs to be addressed in the future, perhaps by choosing a well-distributed dataset, using a more efficient approach for balancing minority classes than the current method, or optimizing the tuning of the classifier's hyperparameters.

We contrast our proposal's performance with earlier works in Tables 8 and 9 that follow. From these two tables, it can be observed that the presented model is superior to competing methods, particularly in terms of the ACC, DR, and FAR, which highlights the effectiveness of our feature selection approach. The performance measures are significantly enhanced by the feature selection methods. During the validation of our method, two benchmark datasets were examined, which involved common threats. While it is true that the other studies may have used more complex algorithms for their experiments, it is important to consider the trade-off between complexity and interpretability when selecting the models for experiments. Sometimes, simpler algorithms can perform just as well or even better than more complex ones. Remember that the performance is not solely dependent on the classifier's or model's complexity but on other factors such as the size of the dataset, its preparation manner, the strategy of the reduction in its dimensions, the extent of the training set balance, the available computational resources, and so on. The proposed approach has achieved better results due to the effectiveness of the feature selection phase, which is a crucial step in building any efficient machine learning model and is preceded by a good preprocessing of the datasets, followed by balancing the training datasets. Additionally, random forest is a powerful and widely used classifier that has been shown to perform well in various applications.

In Table 8, we note that while our approach is generally superior to most other methodologies, study No. 3 is superior in terms of the DR and F-measure, approach No. 13 exceeds based on the DR and FAR, research No. 29 is better in regard to the precision, work No. 32 is outstanding in terms of its accuracy and FAR, and, finally, proposal No. 33 excels in almost everything except the FAR, knowing that it used a deep learning model and took (27,123.87 s) for training and testing. On the contrary, through Table 9, we found an overall superiority of our proposed approach over other methodologies.

We wish to clarify that, while we do not claim that our results are the best ever achieved, our research demonstrates improved performance in most measures compared to the works cited in our study. While we acknowledge that our study employs a standard RF classifier and combines existing techniques for feature selection, we would like to draw attention to the fact that the related work and the methods presented in Tables 8 and 9 do not include unfamiliar or innovative techniques either. However, they have contributed to the body of knowledge in the IDS field. Moreover, some of the cited studies have used the RF classifier, which has shown varying performance. The reason behind the differences in performance lies in the various ways of preprocessing the dataset and the approaches used for selecting features. Instead, we believe that the novelty of our work lies in the integration of these methods (IG, CS, and PSO) to select optimal features carefully after the datasets have been well-prepared and balanced. We hope this clarifies the value and relevance of our research.

Table 14 provides the abbreviations found in Tables 8 and 9, whose full forms are not mentioned in the related works section.

The confusion matrix for our proposition is depicted in Figures 7 and 8. The confusion matrix is a widely used tool in machine learning that provides a comprehensive evaluation of the performance of a model based on a given dataset. Additionally, it helps identify the strengths and weaknesses of the ML model. For example, it can reveal in which classes it performs well and which ones it fails.

Table 10 depicts the evaluation metrics for each type in the UNSW-NB15 dataset, which are derived in dependence on Figure 7, where we can note an outstanding performance, especially in terms of the ACC, FAR, and DR, which reflects

TABLE 9. Based on the Kyoto dataset, the suggested approach is compared to traditional techniques.

No	Ref. No. and Year	Feature Selection	Methods	No. Of Features	ACC	P	R	F	FAR
1	[59] 2014	-	CSV-ISVM	-	-	-	90.14	-	2.314
2	[60] 2015	Filtered, correlation, and consistency	OS-ELM	11	96.37	95.80	97.95	96.86	5.76
3	[61] 2018	-	VAE-Pure	-	-	97.50	75.30	85.00	-
4	[62] 2018	BA	ELM	10	97.96	-	98.75	-	2.425
5	[17] 2019	-	DNN 5-layers	-	88.50	91.30	96.40	93.80	-
6	[19] 2019	NBFS	OSVM-PKNN	18	-	56.89	94.75	-	-
7	[31] 2023	IG-GR-CS	SVM	-	96.42	90.53	96.23	92.96	-
Our proposal		**IG-CS-PSO**	**RF**	**10**	**99.25**	**99.27**	**99.25**	**99.26**	**0.008**

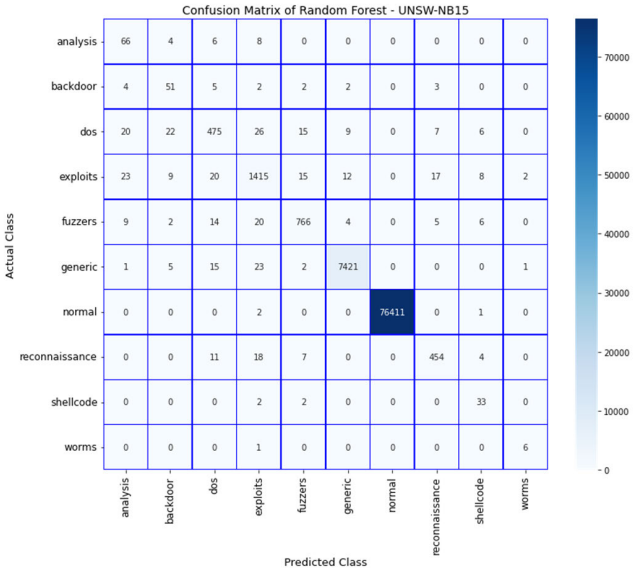

FIGURE 7. Confusion matrix of the UNSW-NB15 dataset.

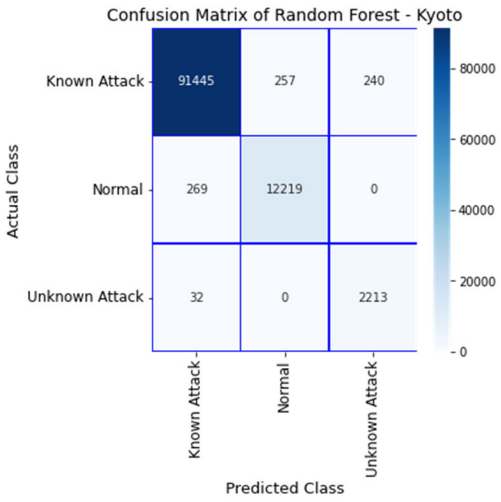

FIGURE 8. Confusion matrix of the Kyoto dataset.

TABLE 10. Evaluation measures for various categories in the UNSW-NB15 dataset.

Category	ACC	P	R	F	FAR
Analysis	99.91	53.66	78.57	63.77	0.065
Backdoor	99.93	54.84	73.91	62.96	0.048
DOS	99.80	87.00	81.90	84.37	0.082
Exploits	99.76	93.28	93.03	93.15	0.119
Fuzzers	99.88	94.68	92.74	93.70	0.050
Generic	99.91	99.64	99.37	99.50	0.034
Normal	1.00	1.00	1.00	1.00	0.000
Reconnaissance	99.92	93.41	91.90	92.65	0.037
Shellcode	99.97	56.90	89.19	69.47	0.028
Worms	1.00	66.67	85.71	75.00	0.003

the robustness of our approach, while both ''normal'' and ''generic'' classes show the best results in all metrics.

Figure 9 and both Tables 11 and 12 demonstrate how our technique has exceeded conventional approaches in terms of accuracy, detection rate, and false alarm rate for the different kinds of categories in the UNSW-NB15 dataset. Most malicious types were identified, along with lowered FAR and increased DR.

Figure 9 displays the power of our approach compared to ref. No. [17] in terms of accuracy in all classes except the ''Fuzzers'' type in the UNSW-NB15 dataset.

According to the UNSW-NB15 dataset, Table 11 displays the effectiveness of our strategy in detecting various types of attacks, except for the ''DOS'' and ''Exploits'' categories in reference No. [43] and the ''Worms'' type in reference No. [47].

While Table 12 illustrates that our strategy has a satisfactory false alarm rate, we observed some superiority of the two references [44] and [17] in certain classes, but overall, our proposal still outperforms them in other measures.

Table 13 shows the evaluation criteria for each kind in the Kyoto dataset, which are generated according to Figure 8, where we can remark on the distinguished results.

Figures 10 and 11 show how our method has outperformed ref No. [31] in terms of the detection rate and false alarm rate for all types of categories in the Kyoto dataset except the ''Unknown Attack'' class concerning the detection rate.

TABLE 11. Performance comparison between the past studies and our proposal based on the DR for each class in the UNSW-NB15.

Class	[43] DR, 2017	[44] DR, 2017	[47] DR, 2018	[16] DR, 2019	[17] DR, 2019	[55] DR, 2020	Our Proposal
							Detection Rate
Normal	97.38	96.30	70.30	82.00	94.70	100.00	100.00
Analysis	69.83	00.90	17.40	1.34	00.00	12.13	78.57
Backdoor	70.44	02.10	16.00	00.00	33.55	63.94	73.91
DOS	84.81	35.70	69.30	0.44	97.80	12.63	81.90
Exploits	95.61	72.80	60.70	57.14	00.17	89.44	93.03
Fuzzers	87.50	28.90	60.70	40.30	00.00	83.86	92.74
Generic	97.81	97.60	96.50	61.21	57.70	97.33	99.37
Reconnaissance	83.80	80.60	83.70	24.89	04.50	66.61	91.90
Shellcode	58.20	29.10	69.30	00.85	00.00	36.51	89.19
Worms	38.24	75.00	90.90	00.00	00.00	24.64	85.71

TABLE 12. Performance comparison between the past studies and our proposal based on the FAR for each class in the UNSW-NB15.

Class	[44] FAR, 2017	[16] FAR, 2019	[17] FAR, 2019	Our Proposal
				False Alarm Rate
Normal	0.106	0.001	0.299	0.000
Analysis	0.000	0.789	0.000	0.065
Backdoor	0.000	0.822	0.000	0.048
DOS	0.023	5.478	0.000	0.082
Exploits	0.073	1.391	0.000	0.119
Fuzzers	0.017	1.321	0.000	0.050
Generic	0.005	0.519	0.155	0.034
Reconnaissance	0.004	1.438	0.000	0.037
Shellcode	0.002	0.435	0.000	0.028
Worms	0.000	0.062	0.000	0.003

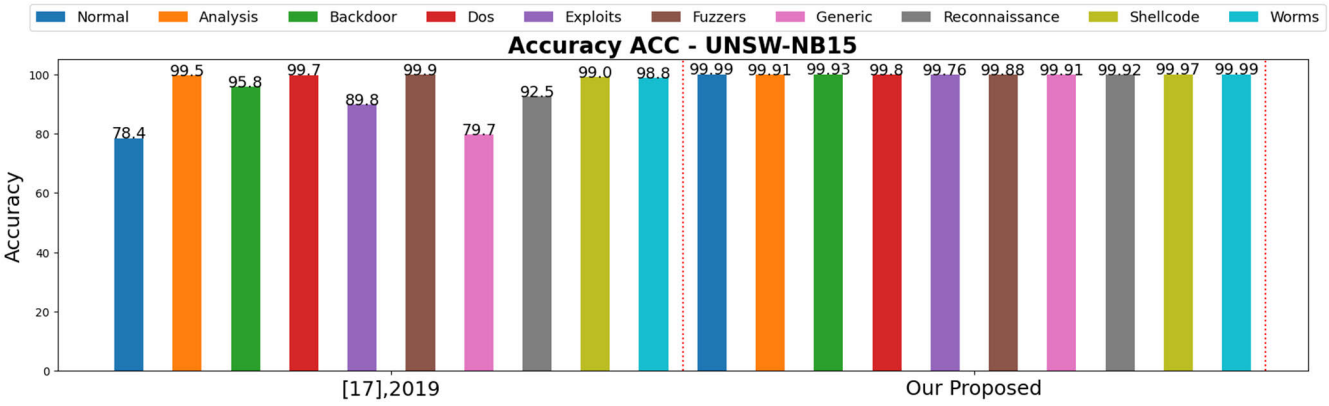

FIGURE 9. Performance comparison between ref No. [17] and our proposal based on the ACC for each class in the UNSW-NB15.

Thus, we can note the importance of PSO in enhancing the results by determining the optimal features, whereas in Ref No. [31], they used IG and CS without any optimization technique.

Figures 12 and 13 present the ROC curve for the multi-class classification, which was plotted based on the values of the FPR and TPR for each class in both Tables 10 and 13. The TPR is known as the recall (R) or the DR, whereas the FPR is also known as the FAR.

The ROC curve can provide valuable insights into the performance of the approach in each category. For example, in the UNSW-NB15 dataset, the "backdoor" class showed the lowest performance, while in the Kyoto dataset, the "normal" type had the lowest performance. Furthermore, we can

TABLE 13. Evaluation measures for various categories in the Kyoto dataset.

Category	ACC	P	R	F	FAR
Normal	99.51	97.94	97.84	97.89	0.003
Known attack	99.25	99.67	99.46	99.56	0.020
Unknown attack	99.74	90.22	98.57	94.21	0.002

TABLE 14. Abbreviation table.

Nomenclature	Abbreviation
SaaS	Software as a Service
PaaS	Platform as a Service
IaaS	Infrastructure as a Service
MLP	Multilayer Perceptron
AlexNet	It's type of DCNN: Deep Convolutional Neural Network
ID3	Iterative Dichotomiser 3
CART	Classification And Regression Tree
LR	Logistic Regression
A-PCA	Adaptive - Principal Component Analysis
GMM	Gaussian Mixture Models
Ramp-KSVCR	Ramp Loss K-Support Vector Classification-Regression
Weka-ML	It's open-source ML software used for feature selection
GA	Genetic Algorithm
GAA-ADS	Geometric Area Analysis-Anomaly-based Detection
RFE-CS	Recursive Feature Elimination - Chi Square
CFS	Correlation-based Feature Selection
ABC-AFS	Artificial Bee Colony-Artificial Fish Swarm
DBN (FE)	Deep Belief Networks (Feature Extraction)
FI	Feature Importance model
FSFF	Feature Similarity-based Fitness Function
CAFF	Classifier Accuracy based Fitness Function
I-ELM	Incremental-Extreme Learning Machine
NSGA2	Non-Dominated Sorting Genetic Algorithm 2
MLR	Multinomial LR
NB-Tree	Naive Bayes Tree
WFEU	Wrapper-based Feature Extraction Unit
FFDNN	Feed-Forward Deep Neural Network
MFFSEM	Multi-dimensional Feature Fusion and Stacking Ensemble Mechanism
TS	Tabu Search
CSV	Candidate Support Vectors
ISVM	Incremental Support Vector Machine
OS-ELM	Online Sequential - ELM
VAE	Variational AutoEncoder
BA	Bat Algorithm
GR	Gain Ratio

obtain an overview of the classifier's overall performance across all classes using a macro-averaged ROC curve.

Therefore, due to the thorough preprocessing of the datasets, which included addressing the imbalance problem and selecting the optimal essential related features using our feature selection approach, as well as the high performance of the random forest classifier, the proposed system showed

TABLE 15. Data type for each feature in the UNSW-NB15 dataset.

No	Feature Name	Type	No	Feature Name	Type
1	srcip	object	26	res_bdy_len	int64
2	sport	int64	27	sjit	float64
3	dstip	object	28	djit	float64
4	dsport	int64	29	stime	int64
5	proto	object	30	ltime	int64
6	state	object	31	sintpkt	float64
7	dur	float64	32	dintpkt	float64
8	sbytes	int64	33	tcprtt	float64
9	dbytes	int64	34	synack	float64
10	sttl	int64	35	ackdat	float64
11	dttl	int64	36	is_sm_ips_ports	int64
12	sloss	int64	37	ct_state_ttl	int64
13	dloss	int64	38	ct_flw_http_mthd	float64
14	service	object	39	is_ftp_login	int64
15	sload	float64	40	ct_ftp_cmd	int64
16	dload	float64	41	ct_srv_src	int64
17	spkts	int64	42	ct_srv_dst	int64
18	dpkts	int64	43	ct_dst_ltm	int64
19	swin	int64	44	ct_src_ltm	int64
20	dwin	int64	45	ct_src_dport_ltm	int64
21	stcpb	int64	46	ct_dst_sport_ltm	int64
22	dtcpb	int64	47	ct_dst_src_ltm	int64
23	smeansz	int64	48	attack_cat	object
24	dmeansz	int64	49	label	int64
25	trans_depth	int64			

TABLE 16. Data type for each feature in the Kyoto dataset.

No	Feature Name	Type	No	Feature Name	Type
1	Duration	float64	13	Dst_host_srv_serror_rate	float64
2	Service	object	14	Flag	object
3	Source bytes	int64	15	IDS_detection	object
4	Destination bytes	int64	16	Malware_detection	object
5	Count	int64	17	Ashula_detection	object
6	Same_srv_rate	float64	18	Label	int64
7	Serror_rate	float64	19	Source_IP_Address	object
8	Srv_serror_rate	float64	20	Source_Port_Number	int64
9	Dst_host_count	int64	21	Destination_IP_Address	object
10	Dst_host_srv_count	int64	22	Destination_Port_Number	int64
11	Dst_host_same_src_port_rate	float64	23	Start_Time	object
12	Dst_host_serror_rate	float64	24	Protocol	object

greater effectiveness when compared with other existing techniques. Our suggested system achieved a high DR and a low FAR, demonstrating its efficacy in detecting intrusions within cloud environments. The results indicate that our system can

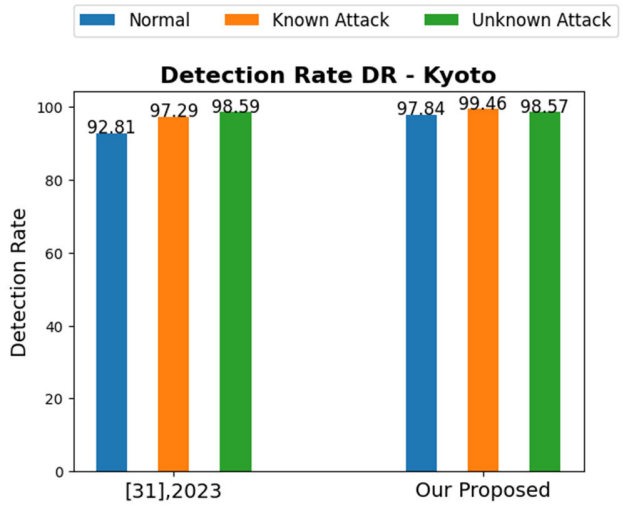

FIGURE 10. Performance comparison between ref No. [31] and our proposal based on the DR for each class in the Kyoto.

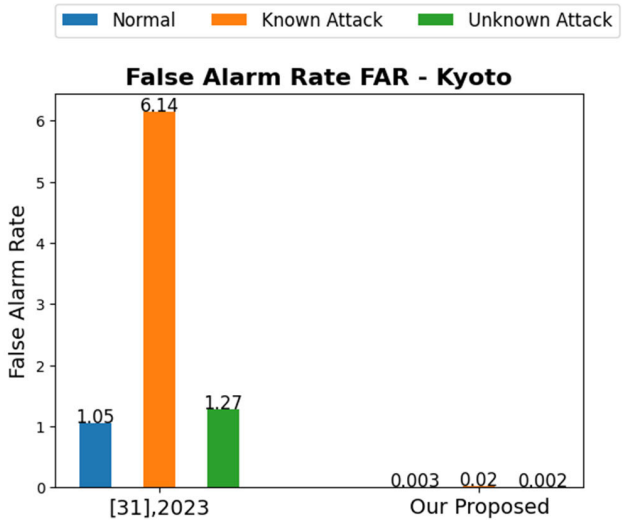

FIGURE 11. Performance comparison between ref No. [31] and our proposal based on the FAR for each class in the Kyoto.

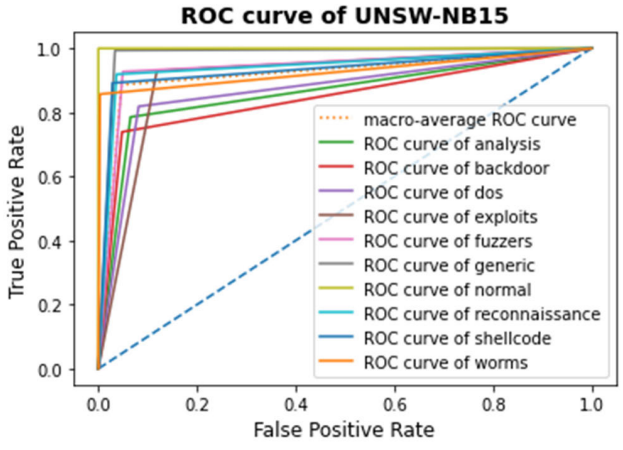

FIGURE 12. ROC curve of the UNSW-NB15 dataset.

successfully identify potential cloud attack types, such as those mentioned in the utilized datasets. Consequently, it is well-suited for deployment in cloud conditions.

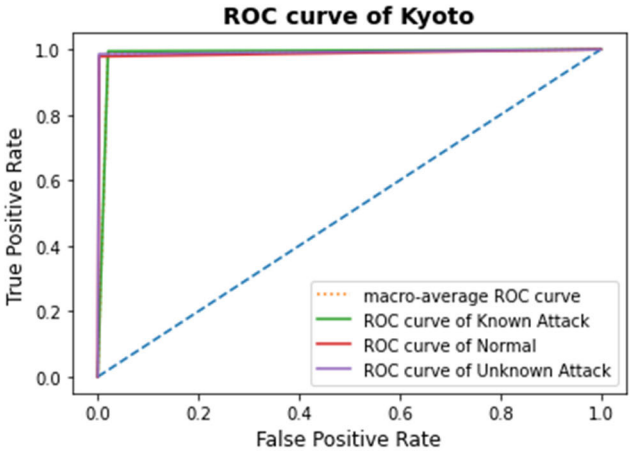

FIGURE 13. ROC curve of the Kyoto dataset.

VI. CONCLUSION

With the widespread use of cloud computing by people and businesses, security in the cloud is of the utmost concern. Thus, the potential of machine learning models to classify incoming network packets as normal or abnormal was exploited to identify intrusions and preserve user data, and this was done at an acceptable resource cost that distinguishes machine learning models from deep learning models. The suggested intrusion detection system aims to develop a model that would leverage increased intrusion detection's accuracy by combining the strengths of each employed feature selection algorithm (information gain, chi-square, and particle swarm optimization) to find an optimal feature subset that not only enhances the performance of the model but also offers useful features that are strongly associated with the target variable. The proposal displayed its power to identify multiple attack types with a greater DR and a lower FAR.

In the upcoming work, we plan to use deep learning techniques to obtain high performance, along with ensemble learning concepts and other meta-heuristic optimization algorithms; also, the testing will be carried out through the most recent datasets, which contain a broad range of threats that accurately simulate the real networks of today.

APPENDIX A

Table 14 presents the abbreviations used, which have not been mentioned in their full forms throughout the manuscript. They are listed in the order of appearance.

APPENDIX B

Tables 15 and 16 represent the data type for each feature in the UNSW-NB15 and Kyoto datasets, respectively. These data types have been determined using the info() method provided by the Pandas library.

ACKNOWLEDGMENT

The authors would like to acknowledge the Researchers Supporting Project number (RSP2023R476), King Saud University, Riyadh, Saudi Arabia.

REFERENCES

[1] R. R. Kumar, A. Tomar, M. Shameem, and M. N. Alam, "OPTCLOUD: An optimal cloud service selection framework using QoS correlation lens," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–16, May 2022, doi: 10.1155/2022/2019485.

[2] R. R. Kumar, M. Shameem, R. Khanam, and C. Kumar, "A hybrid evaluation framework for QoS based service selection and ranking in cloud environment," in *Proc. 15th IEEE India Council Int. Conf.*, Oct. 2018, pp. 1–6, doi: 10.1109/INDICON45594.2018.8987192.

[3] M. Bakro, S. K. Bisoy, A. K. Patel, and M. A. Naal, "Performance analysis of cloud computing encryption algorithms," in *Advances in Intelligent Computing and Communication*, in Lecture Notes in Networks and Systems, vol. 202. Singapore: Springer, 2021, pp. 357–367, doi: 10.1007/978-981-16-0695-3_35.

[4] (2020). *Malware Statistics & Trends Report | AV-TEST*. Accessed: Jan. 21, 2023. [Online]. Available: https://www.av-test.org/en/statistics/malware/

[5] *Digital Technology Market Research Services | Juniper Research*. Accessed: Jan. 21, 2023. [Online]. Available: https://www.juniperresearch.com/home

[6] *Cyber Security Market Size, Share & Trends Report, 2030*. Accessed: Jan. 21, 2023. [Online]. Available: https://www.grandviewresearch.com/industry-analysis/cyber-security-market

[7] R. R. Kumar, M. Shameem, and C. Kumar, "A computational framework for ranking prediction of cloud services under fuzzy environment," *Enterprise Inf. Syst.*, vol. 16, no. 1, pp. 167–187, Jan. 2022, doi: 10.1080/17517575.2021.1889037.

[8] M. A. Akbar, M. Shameem, S. Mahmood, A. Alsanad, and A. Gumaei, "Prioritization based taxonomy of cloud-based outsource software development challenges: Fuzzy AHP analysis," *Appl. Soft Comput.*, vol. 95, Oct. 2020, Art. no. 106557, doi: 10.1016/j.asoc.2020.106557.

[9] M. Bakro, R. R. Kumar, A. A. Alabrah, Z. Ashraf, S. K. Bisoy, N. Parveen, S. Khawatmi, and A. Abdelsalam, "Efficient intrusion detection system in the cloud using fusion feature selection approaches and an ensemble classifier," *Electronics*, vol. 12, no. 11, p. 2427, May 2023, doi: 10.3390/electronics12112427.

[10] M. Bakro, S. K. Bisoy, A. K. Patel, and M. A. Naal, "Hybrid blockchain-enabled security in cloud storage infrastructure using ECC and AES algorithms," in *Blockchain based Internet of Things*. Singapore: Springer, 2022, pp. 139–170, doi: 10.1007/978-981-16-9260-4_6.

[11] Z. Ahmad, A. S. Khan, C. W. Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 1, p. e4150, Jan. 2021, doi: 10.1002/ett.4150.

[12] I. F. Kilincer, F. Ertam, and A. Sengur, "Machine learning methods for cyber security intrusion detection: Datasets and comparative study," *Comput. Netw.*, vol. 188, Apr. 2021, Art. no. 107840, doi: 10.1016/j.comnet.2021.107840.

[13] I. Benmessahel, K. Xie, and M. Chellal, "A new evolutionary neural networks based on intrusion detection systems using multiverse optimization," *Int. J. Speech Technol.*, vol. 48, no. 8, pp. 2315–2327, Aug. 2018, doi: 10.1007/S10489-017-1085-Y.

[14] Y. Yang, K. Zheng, C. Wu, and Y. Yang, "Improving the classification effectiveness of intrusion detection by using improved conditional variational AutoEncoder and deep neural network," *Sensors*, vol. 19, no. 11, p. 2528, Jun. 2019, doi: 10.3390/s19112528.

[15] B. A. Tama, M. Comuzzi, and K. Rhee, "TSE-IDS: A two-stage classifier ensemble for intelligent anomaly-based intrusion detection system," *IEEE Access*, vol. 7, pp. 94497–94507, 2019, doi: 10.1109/ACCESS.2019.2928048.

[16] F. A. Khan, A. Gumaei, A. Derhab, and A. Hussain, "TSDL: A two-stage deep learning model for efficient network intrusion detection," *IEEE Access*, vol. 7, pp. 30373–30385, 2019, doi: 10.1109/ACCESS.2019.2899721.

[17] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019, doi: 10.1109/ACCESS.2019.2895334.

[18] R. Patil, H. Dudeja, and C. Modi, "Designing an efficient security framework for detecting intrusions in virtual network of cloud computing," *Comput. Secur.*, vol. 85, pp. 402–422, Aug. 2019, doi: 10.1016/j.cose.2019.05.016.

[19] A. I. Saleh, F. M. Talaat, and L. M. Labib, "A hybrid intrusion detection system (HIDS) based on prioritized k-nearest neighbors and optimized SVM classifiers," *Artif. Intell. Rev.*, vol. 51, no. 3, pp. 403–443, Mar. 2019, doi: 10.1007/s10462-017-9567-1.

[20] J. Zhang, Y. Ling, X. Fu, X. Yang, G. Xiong, and R. Zhang, "Model of the intrusion detection system based on the integration of spatial–temporal features," *Comput. Secur.*, vol. 89, Feb. 2020, Art. no. 101681, doi: 10.1016/j.cose.2019.101681.

[21] S. M. Kasongo and Y. Sun, "Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset," *J. Big Data*, vol. 7, no. 1, pp. 1–12, Dec. 2020, doi: 10.1186/s40537-020-00379-6.

[22] V. Kumar, D. Sinha, A. K. Das, S. C. Pandey, and R. T. Goswami, "An integrated rule based intrusion detection system: Analysis on UNSW-NB15 data set and the real time online dataset," *Cluster Comput.*, vol. 23, no. 2, pp. 1397–1418, Jun. 2020, doi: 10.1007/s10586-019-03008-x.

[23] O. Almomani, "A feature selection model for network intrusion detection system based on PSO, GWO, FFA and GA algorithms," *Symmetry*, vol. 12, no. 6, pp. 1–20, 2020, doi: 10.3390/sym12061046.

[24] K. Jiang, W. Wang, A. Wang, and H. Wu, "Network intrusion detection combined hybrid sampling with deep hierarchical network," *IEEE Access*, vol. 8, pp. 32464–32476, 2020, doi: 10.1109/ACCESS.2020.2973730.

[25] P. Rajesh Kanna and P. Santhi, "Unified deep learning approach for efficient intrusion detection system using integrated spatial–temporal features," *Knowl.-Based Syst.*, vol. 226, Aug. 2021, Art. no. 107132, doi: 10.1016/j.knosys.2021.107132.

[26] G. Sreelatha, A. V. Babu, and D. Midhunchakkaravarthy, "Improved security in cloud using sandpiper and extended equilibrium deep transfer learning based intrusion detection," *Cluster Comput.*, vol. 25, no. 5, pp. 3129–3144, Oct. 2022, doi: 10.1007/s10586-021-03516-9.

[27] P. R. Kanna and P. Santhi, "Hybrid intrusion detection using MapReduce based black widow optimized convolutional long short-term memory neural networks," *Expert Syst. Appl.*, vol. 194, May 2022, Art. no. 116545, doi: 10.1016/j.eswa.2022.116545.

[28] S. Krishnaveni, S. Sivamohan, S. S. Sridhar, and S. Prabakaran, "Efficient feature selection and classification through ensemble method for network intrusion detection on cloud computing," *Cluster Comput.*, vol. 24, no. 3, pp. 1761–1779, Sep. 2021, doi: 10.1007/s10586-020-03222-y.

[29] K. Potdar, "A comparative study of categorical variable encoding techniques for neural network classifiers," *Int. J. Comput. Appl.*, vol. 175, no. 4, pp. 7–9, Oct. 2017, doi: 10.5120/ijca2017915495.

[30] M. Rashid, J. Kamruzzaman, T. Imam, S. Wibowo, and S. Gordon, "A tree-based stacking ensemble technique with feature selection for network intrusion detection," *Int. J. Speech Technol.*, vol. 52, no. 9, pp. 9768–9781, Jul. 2022, doi: 10.1007/s10489-021-02968-1.

[31] M. Bakro, R. R. Kumar, S. K. Bisoy, M. O. Addas, and D. Khamis, "Developing a cloud intrusion detection system with filter-based features selection techniques and SVM classifier," in *Proc. Int. Conf. Comput., Commun. Learn.*, vol. 1729. Cham, Switzerland: Springer, 2023, pp. 15–26, doi: 10.1007/978-3-031-21750-0_2.

[32] N. Arora and P. D. Kaur, "A bolasso based consistent feature selection enabled random forest classification algorithm: An application to credit risk assessment," *Appl. Soft Comput.*, vol. 86, Jan. 2020, Art. no. 105936, doi: 10.1016/j.asoc.2019.105936.

[33] M. Shameem and M. Nadeem, "Genetic algorithm based probabilistic model for agile project success in global software development," *SSRN Electron. J.*, vol. 2022, pp. 1–10, Jan. 2022, doi: 10.2139/ssrn.4115147.

[34] P. Wei, Y. Li, Z. Zhang, T. Hu, Z. Li, and D. Liu, "An optimization method for intrusion detection classification model based on deep belief network," *IEEE Access*, vol. 7, pp. 87593–87605, 2019, doi: 10.1109/ACCESS.2019.2925828.

[35] M. H. Nasir, S. A. Khan, M. M. Khan, and M. Fatima, "Swarm intelligence inspired intrusion detection systems—A systematic literature review," *Comput. Netw.*, vol. 205, Mar. 2022, Art. no. 108708, doi: 10.1016/j.comnet.2021.108708.

[36] C. Khammassi and S. Krichen, "A GA-LR wrapper approach for feature selection in network intrusion detection," *Comput. Secur.*, vol. 70, pp. 255–277, Sep. 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404817301244

[37] 2015. *The UNSW-NB15 Dataset | UNSW Research*. Accessed: Feb. 14, 2023. [Online]. Available: https://research.unsw.edu.au/projects/unsw-nb15-dataset

[38] Kyoto University. (2006). *Traffic Data From Kyoto University's Honeypots*. Accessed: Feb. 14, 2023. [Online]. Available: http://www.takakura.com/kyoto_data/

[39] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Mil. Commun. Inf. Syst. Conf.*, 2015, pp. 1–6, doi: 10.1109/MilCIS.2015.7348942.

[40] J. Song, H. Takakura, Y. Okabe, M. Eto, D. Inoue, and K. Nakao, "Statistical analysis of honeypot data and building of Kyoto 2006+ dataset for NIDS evaluation," in *Proc. 1st Workshop Building Anal. Datasets Gathering Exper. Returns Secur.*, Apr. 2011, pp. 29–36, doi: 10.1145/1978672.1978676.

[41] N. Moustafa and J. Slay, "The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Inf. Secur. J., Global Perspective*, vol. 25, nos. 1–3, pp. 18–31, Apr. 2016, doi: 10.1080/19393555.2015.1125974.

[42] N. Moustafa, G. Creech, E. Sitnikova, and M. Keshk, "Collaborative anomaly detection framework for handling big data of cloud computing," in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, Nov. 2017, pp. 1–6, doi: 10.1109/MilCIS.2017.8190421.

[43] S. M. Hosseini Bamakan, H. Wang, and Y. Shi, "Ramp loss K-support vector classification-regression; a robust and sparse multi-class approach to the intrusion detection problem," *Knowl.-Based Syst.*, vol. 126, pp. 113–126, Jun. 2017, doi: 10.1016/j.knosys.2017.03.012.

[44] T. Janarthanan and S. Zargari, "Feature selection in UNSW-NB15 and KDDCUP'99 datasets," in *Proc. IEEE 26th Int. Symp. Ind. Electron. (ISIE)*, Jun. 2017, pp. 1881–1886, doi: 10.1109/ISIE.2017.8001537.

[45] N. Moustafa, J. Slay, and G. Creech, "Novel geometric area analysis technique for anomaly detection using trapezoidal area estimation on large-scale networks," *IEEE Trans. Big Data*, vol. 5, no. 4, pp. 481–494, Dec. 2019, doi: 10.1109/TBDATA.2017.2715166.

[46] P. Mishra, E. S. Pilli, V. Varadharajan, and U. Tupakula, "Out-VM monitoring for malicious network packet detection in cloud," in *Proc. ISEA Asia Secur. Privacy (ISEASP)*, Jan. 2017, pp. 1–4, doi: 10.1109/ISEASP.2017.7976995.

[47] W. Zong, Y. W. Chow, and W. Susilo, "A two-stage classifier approach for network intrusion detection," in *Proc. Int. Conf. Inf. Secur. Pract. Exper.*, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 11125, 2018, pp. 329–340, doi: 10.1007/978-3-319-99807-7_20.

[48] V. Hajisalem and S. Babaie, "A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection," *Comput. Netw.*, vol. 136, pp. 37–50, May 2018, doi: 10.1016/j.comnet.2018.02.028.

[49] N. Marir, H. Wang, G. Feng, B. Li, and M. Jia, "Distributed abnormal behavior detection approach based on deep belief network and ensemble SVM using spark," *IEEE Access*, vol. 6, pp. 59657–59671, 2018, doi: 10.1109/ACCESS.2018.2875045.

[50] P. Dahiya and D. K. Srivastava, "A comparative evolution of unsupervised techniques for effective network intrusion detection in Hadoop," in *Proc. Int. Conf. Adv. Comput. Data Sci.*, vol. 906, 2018, pp. 279–287, doi: 10.1007/978-981-13-1813-9_28.

[51] N. M. Khan, N. C. Madhav, A. Negi, and I. S. Thaseen, "Analysis on improving the performance of machine learning models using feature selection technique," in *Proc. Int. Conf. Intell. Syst. Design Appl.*, in Advances in Intelligent Systems and Computing, vol. 941, 2018, pp. 69–77, doi: 10.1007/978-3-030-16660-1_7.

[52] J. Gao, S. Chai, B. Zhang, and Y. Xia, "Research on network intrusion detection based on incremental extreme learning machine and adaptive principal component analysis," *Energies*, vol. 12, no. 7, p. 1223, Mar. 2019, doi: 10.3390/en12071223.

[53] A. S. Almogren, "Intrusion detection in edge-of-things computing," *J. Parallel Distrib. Comput.*, vol. 137, pp. 259–265, Mar. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S074373151930872X

[54] C. Khammassi and S. Krichen, "A NSGA2-LR wrapper approach for feature selection in network intrusion detection," *Comput. Netw.*, vol. 172, May 2020, Art. no. 107183, doi: 10.1016/j.comnet.2020.107183.

[55] I. Sumaiya Thaseen, J. Saira Banu, K. Lavanya, M. Rukunuddin Ghalib, and K. Abhishek, "An integrated intrusion detection system using correlation-based attribute selection and artificial neural network," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 2, pp. 1–15, Feb. 2021, doi: 10.1002/ett.4014.

[56] S. M. Kasongo and Y. Sun, "A deep learning method with wrapper based feature extraction for wireless intrusion detection system," *Comput. Secur.*, vol. 92, May 2020, Art. no. 101752, doi: 10.1016/j.cose.2020.101752.

[57] H. Zhang, J.-L. Li, X.-M. Liu, and C. Dong, "Multi-dimensional feature fusion and stacking ensemble mechanism for network intrusion detection," *Future Gener. Comput. Syst.*, vol. 122, pp. 130–143, Sep. 2021, doi: 10.1016/j.future.2021.03.024.

[58] A. Nazir and R. A. Khan, "A novel combinatorial optimization based feature selection method for network intrusion detection," *Comput. Secur.*, vol. 102, Mar. 2021, Art. no. 102164. https://www.sciencedirect.com/science/article/pii/S0167404820304375.

[59] R. Chitrakar and C. Huang, "Selection of candidate support vectors in incremental SVM for network intrusion detection," *Comput. Secur.*, vol. 45, pp. 231–241, Sep. 2014, doi: 10.1016/j.cose.2014.06.006.

[60] R. Singh, H. Kumar, and R. K. Singla, "An intrusion detection system using network traffic profiling and online sequential extreme learning machine," *Expert Syst. Appl.*, vol. 42, no. 22, pp. 8609–8624, Dec. 2015, doi: 10.1016/j.eswa.2015.07.015.

[61] R. K. Malaiya, D. Kwon, S. C. Suh, H. Kim, I. Kim, and J. Kim, "An empirical evaluation of deep learning for network anomaly detection," *IEEE Access*, vol. 7, pp. 140806–140817, 2019, doi: 10.1109/ACCESS.2019.2943249.

[62] Y. Shen, K. Zheng, C. Wu, M. Zhang, X. Niu, and Y. Yang, "An ensemble method based on selection using bat algorithm for intrusion detection," *Comput. J.*, vol. 61, no. 4, pp. 526–538, Apr. 2018, doi: 10.1093/comjnl/bxx101.

MHAMAD BAKRO received the M.Tech. degree (Hons.) in computer science and engineering from C. V. Raman Global University, Bhubaneswar, India, in 2020 (with a final grade of 85.10%), where he is currently pursuing the Ph.D. degree in computer science and engineering. He has more than three years of teaching experience at C. V. Raman Global University. He has published three conferences, one book chapter, and one journal article. His research interests include cloud computing, intrusion detection systems, machine learning, deep learning, and optimization algorithms. He is a reviewer for several journals and conferences.

RAKESH RANJAN KUMAR received the M.Tech. degree from MNNIT, Allahabad, India, and the Ph.D. degree from IIT (ISM), Dhanbad, India. He is currently working as an Assistant Professor with the Department of CSE, C. V. Raman Global University, India. He has published more than 15 papers in reputed journals and conferences. His current research interests include cloud computing, service selection, and optimization. He acted as a reviewer in many reputed journals and conferences.

AMERAH ALABRAH received the M.S. degree in computer science from Colorado State University, in 2008, and the Ph.D. degree in computer science from the College of Computer Science and Engineering, University of Central Florida, in 2014. Her research is mainly focused in computer and network security and more specifically in optimizing security measures for social media networks. She is currently working as an Associate Professor at the College of Computer and Information Sciences, King Saud University, and a member of the Saudi Telecom Company Artificial Intelligence Research Fund.

ZUBAIR ASHRAF received the Ph.D. degree in computer science from South Asian University, New Delhi, India, in February 2020. He is currently an Assistant Professor with the Department of Computer Engineering and Applications, GLA University, Uttar Pradesh, India. His research interests include machine learning and optimization, nature-inspired intelligent computation, deep learning, and fuzzy systems. He is also an active Reviewer of journals such as IEEE TRANSACTIONS ON FUZZY SYSTEMS, *Soft Computing*, *Applied Soft Computing*, *Journal of Applied Mathematics*, and the *International Journal of Intelligent Systems*.

MD NADEEM AHMED is currently working for the Department of Computer Science Engineering, Chandigarh University, as an Assistant Professor. He has published several papers in Scopus, SCI-indexed journals, and conferences. He is having around seven years of experience in teaching and industry. He is cloud-certified, java certified, oracle developer-certified, and big data certified. His research interests include big data, machine learning, object-oriented programming, and cloud computing.

MOHAMMAD SHAMEEM received the Ph.D. degree from the Indian Institute of Technology (Indian School of Mines), Dhanbad. Currently, he is working as an Assistant Professor with the Department of Computer Science and Engineering, KLEF deemed to be University, Guntur, AP, India. He has published various papers in well-reputed SCI and Scopus journals, i.e., *Journal of Software Evolution and Process* (Wiley), *Applied Soft Computing* (Elsevier), and *Arabian Journal of Science and Engineering* (Springer). Moreover, he has presented his research papers in various international and national conferences such as APSEC, SKIMA, EASE, and QSE. His research interests include agile software development, empirical software engineering, global software engineering, machine learning, and optimization.

AHMED ABDELSALAM received the B.Sc. degree in mechatronics and robotics from the Egyptian Russian University, Egypt, in 2015, the double M.Sc. degree in mechatronics and robotics from ITMO University, Russia, and in mechanical engineering from Lappeenranta-Lahti University of Technology, Finland, in 2019. He is currently pursuing the Ph.D. degree with the Software Department, Lappeenranta-Lahti University of Technology, researching the field of semantic segmentation, machine learning, and autonomous navigation.