Intrusion Detection Technique in Wireless Sensor Network using Grid Search Random Forest with Boruta Feature Selection Algorithm

Sridevi Subbiah, Kalaiarasi Sonai Muthu Anbananthen, Saranya Thangaraj, Subarmaniam Kannan, and Deisy Chelliah

***Abstract*—Attacks in wireless sensor networks (WSNs) aim to prevent or eradicate the network's ability to perform its anticipated functions. Intrusion detection is a defense used in wireless sensor networks that can detect unknown attacks. Due to the incredible development in computer-related applications and massive Internet usage, it is indispensable to provide host and network security. The development of hacking technology tries to compromise computer security through intrusion. Intrusion detection system (IDS) was employed with the help of machine learning (ML) Algorithms to detect intrusions in the network. Classic ML algorithms like support vector machine (SVM), K-nearest neighbour (KNN), and filter-based feature selection often led to poor accuracy and misclassification of intrusions. This article proposes a novel framework for IDS that can be enabled by Boruta feature selection with grid search random forest (BFS-GSRF) algorithm to overcome these issues. The performance of BFS-GSRF is compared with ML algorithms like linear discriminant analysis (LDA) and classification and regression tree (CART) etc. The proposed work was implemented and tested on network security laboratory – knowledge on discovery dataset (NSL-KDD). The experimental results show that the proposed model BFS-GSRF yields higher accuracy (i.e., 99%) in detecting attacks, and it is superior to LDA, CART, and other existing algorithms.**

***Index Terms*—Boruta feature selection, grid search random forest, intrusion detection system (IDS), machine learning (ML), wireless sensor networks (WSNs).**

I. Introduction

APPLICATIONS in industry, research, business, healthcare, and human lives depend heavily on wired & wireless computer networks, and their valuable information is transferred all over the Internet every second. Therefore, the data needs to be protected against intruders. The attackers mainly focus on acquiring, destroying, and modifying the most valuable information to attain financial gain or risk the target host or network. Wireless sensor networks (WSNs) are more vulnerable [1] such as open-air transmission, dynamic network topology, broadcasting medium, constrained node network, and insufficient physical infrastructure. This vulnerability brings various security attacks [2]. The WSNs are usually built in the unattended environment of open-air communication, making WSNs more prone to cyber-attacks than a wired network. All the nodes in the WSNs, communicate with each other; if any one of the nodes got compromised by the attacker, then the entire WSNs system will lead to misleading communication information. The most common attacks in WSNs are jamming, spoofing, hijacking, and eavesdropping [3]. The WSNs connected to the IoT gadgets like sensors, actuators and, other wireless devices need an intrusion detection system with the optimized method to identify abnormal behaviours.

Intrusion detection system (IDS) is one of the refuge skills developed for detecting unguarded things which are very vulnerable in a host or network. An IDS is an application or hardware device that will observe the host or networks exposed to the attacks and create alerts and warnings to the admin or Security professional. The security professionals are responsible to heed the notifications generated by IDS. This IDS is also extended to heed the warnings by blocking suspicious activity, and threats breaching the policy without security professionals are called IDS/IPS intrusion prevention system (IPS). The IDS is a listen-only device, where it just keeps on monitoring host and network for malicious activity and detects once such an activity is called passive IDS. If it is preventing malicious activity, then it is called active IDS.

Based on the deployment of IDS, it is categorized into (i) host intrusion detection system (HIDS) and (ii) network intrusion detection system (NIDS). The host-based IDS are IDS specially designed to examine independent system actions such as monitoring log files. HIDS will inspect the incoming and outgoing packets for the system where it is deployed. It also monitors the operating system (OS) of the host system. The HIDS will snap a picture of the entire file system set and compare it with the file system's previous image. From the comparison result, if there are any changes or modifications in the file system, it will alert the admin. The tool used for HIDS is OSSEC and Sagan etc. The network-based IDS is designed to examine network traffic from the entire host associated with the network. The traffic is analyzed on a whole subnet and compared with the traffic previously passed by those attacks. If the comparison discovers any kinds of threats, it will generate a warning or alert to the network admin. The tools mainly used for NIDS are Snort and Bro etc. The main difference between HIDS

Manuscript received November 9, 2021; revised January 3, 2022; approved for publication by Junbeom Hur, Division III Editor, January 3, 2022.

Sridevi, S, Saranya, T., and Deisy, C. are with Thiagarajar College of Engineering, Madurai, email: sridevi@tce.edu, saranshakthi09@gmail.com, cdcse@tce.edu.

Kalaiarasi, S.M.A. and Subarmaniam, K. are with Multimedia University, Malaysia, email: {kalaiarasi, subar.kannan}@mmu.edu.my.

Kalaiarasi, S.M.A and Sridevi, S are the corresponding authors.

Digital Object Identifier: 10.23919/JCN.2022.000002

and NIDS is that HIDS will work on the log file system, whereas NIDS will work on live network traffic data.

The detection techniques are classified into three types: Signature-based-detection, anomaly-based-detection, and hybrid-based-detection [4]. An anomaly detection takes the baseline of normal traffic behaviour and computes the present state of network traffic with the baseline. If it varies from the baseline, then it alerts it as an attack to the network admin. The signature-based intrusion detection relies on the database of previous attacks and known susceptibilities of the structure. The attackers leave a footprint at each intrusion is called the signature. This signature is used to identify the same threats which repeat in the future. The signature-based detection is also analyzed based on available traffic data, called knowledge-based detection. Hybrid-based detection combines anomaly-based and signature-based detection to provide better detection at low false-positive rates and high detection rates. It has a high probability of finding unknown threats.

The traditional detection methods are not efficient for detecting intrusions on huge data. Machine learning (ML) algorithms can improve intrusion detection efficiency [5]. ML can be classified into supervised, unsupervised, and semi-supervised types. In a supervised method, the labelled input is given to the system for training. With the help of the label, it will separate the different classes available in the dataset. In an unsupervised method, the unlabeled input is given to the system, which will figure out the structure of similarity presented in the input data. A semi-supervised approach uses a few labelled data with many unlabeled data. This method drops between the supervised and unsupervised methods. The accuracy of semi-supervised learning can be improved by using both labelled and unlabeled data.

In the existing literature (Zhicong *et al.*, (2018) [6]; Abhale & Manivannan (2020) [7]; Saranya *et al.*, (2020)) [8], the authors' used algorithms like logistic regression, K-nearest neighbours (KNN), support vector machines (SVM), artificial neural network (ANN), convolution neural network (CNN), and random forest (RF) for intrusion detection [1], [9]. The accuracy of these algorithms ranges from 75% to 90%. The accuracy of the classifier mainly depends on the dataset and feature selection on the dataset. In the above-said algorithms, the feature selection algorithm is not embedded. Hence the accuracy is less than 90%. This research work proposed Boruta feature selection with grid search random forest (BFS-GSRF) algorithm to improve the classifier's performance through the feature selection method.

The remaining section of the article is organized as follows: Section II focuses on related work for intrusion detection, Section III covers dataset details used in the proposed work, Section IV describes the proposed model, Section V focuses on results and discussion and comparison of the proposed model with existing algorithms, and Section VI of the report concludes with conclusions and suggestions for future works.

II. Literature Survey

People's widespread usage of devices and technology creates enormous amounts of data for every second. Security is required for such massive data [10] to secure the host and the data that resides in the host. Machine learning algorithms will employ to ensure those data and the host by detecting intrusions. The machine learning technique makes use of statistics and algorithms to find the intrusion in the networks. Some of the literature that uses machine learning algorithms for IDS are discussed below.

The nodes in WSNs are lightweight and resource-constrained, so the paper [3] proposed a hybrid, lightweight IDS for sensor networks. Cluster-based architecture is used to reduce energy, and the anomalies in sensors are detected by hybridization of SVM and a set of signature rules. The work was tested in a simulation environment.

All the OSI model layers observe the attacks in WSNs [7]. WSNs should employ a monitoring system to identify attacks, especially for security purposes. They experiment with various supervised machine learning algorithms like RF, SVM, decision trees, Ada boost classifier, K nearest neighbour classifier, Gaussian naïve Bayes, and logistic regression classifier. Tested on NSLKDD data set, and their result shows that SVM achieves higher accuracy than existing algorithms.

Logistic regression is used to predict the probability of the target variable for binary classification and multi-classification problems. The likelihood of an event occurring can be anticipated by fitting the data into the logistic function [11]. The sigmoid function is used to map prediction to probabilities. RF is one of the ML algorithms used for applications like stock market prediction, disease classification, fraud detection, etc. The RF builds a different decision tree [12] and gets the prediction from each decision tree [13]. Then it merges the predictions of multiple decision trees to find the final prediction utilizing voting. RF is much better than decision trees because it limits over-fitting.

SVM is an ML algorithm used for classification and regression. SVM [12] uses a linear hyperplane for classification, and it is called linearly separable. K-means clustering is one of the unsupervised learning algorithms used to partition the data into subgroups called a cluster. In this algorithm, each data belongs to one group, and it also creates an inter-cluster by maintaining the cluster [14] as far as possible. The partitioning was done based on the K value, which refers to averaging data points to find its centroid [15]. In the K-means clustering algorithm, the resultant cluster highly depends on the initialization of the parameter. They have used the modified K-means algorithm [16] K-harmonic means (KHM) algorithm to predict time series data.

KNN is another supervised algorithm used for classification and regression. KNN uses some labelled data points to learn how to label other data points with the help of nearest neighbours. The performance of KNN is poorer for unbalanced data, and this was solved in the paper [17] by introducing density into KNN for predicting IDS more precisely. A wireless mesh network [18] is highly subjected to cyber-attacks. The genetic-based feature selection algorithm and multiple support vector

machines [19] classify attacks in wireless mesh networks simulated on network simulator 3. They have compared the proposed algorithm with existing machine learning algorithms. Their proposed algorithms yield better accuracy than existing algorithms.

Logarithm marginal density ratios transformation (LMDRT)-SVM for IDS was introduced in the research work [20]. The work was proposed to improve the SVM detection ratio. They implemented the LMDRT algorithm to extract the exact features. The steps used to build IDS are: Data transformation is done using LMDRT, new data is formed, an SVM classifier is used to train to form a detection model. Then with a new testing sample, the intrusion is detected based on a trained classifier. Later, least square support vector machine (LS-SVM) algorithm was used for intrusion detection systems [21]. It can be applied for both static and incremental data.

Machine learning IDS for mobile cloud was discussed in [22]. Their scheme covers two steps: Multi-layer traffic screening and decision-based virtual machine (VM) selection. The ML algorithm such as Gaussian naïve Bayes, SVM, and RF are tested with NSL-KDD data set [11]. They proved that RF achieves the highest accuracy and outperforms the other approaches.

Lightweight SVM [23] is used for detecting IDS in WSNs IoT networks. It is based on the lightweight concept to utilize the minimum energy and resources available within IoT sensors nodes. IDS is entirely relied on the packet arrival rate and considers the attributes. The author used the MATLAB simulation tool to implement and collect data. They implemented the models such as K-NN, ANN, and SVM. The performance comparison on several machine learning algorithms such as SVM, DT, RF, and ANN [24] was made to predict intrusions on the IoT environment. Their results show that DT, RF, and ANN performed well.

Hypergraph clustering model (HC-IDS) based on Apriori algorithm to detect DDoS attack on fog computing was discussed in [25]. Apriori algorithm describes the association between fog nodes that are affected by DDoS attacks. They tested this work under the simulated environment of a radio communication system and a plurality of fog nodes.

Chiba *et al.* (2019) [22] discussed a hybrid optimization framework for network anomaly intrusion detection using a deep neural network. The work focused on improved genetic algorithm and simulated annealing algorithm (IGASAA). At first, construct the important attributes from the dataset using IGASAA, which helps find the optimal/near-optimal attribute. They used four modules to build MLIDS in optimization mode such as feature selection module: 70 features are selected from 80 features, data pre-processing module: Encoding and normalization are done, detection module: IDS is built based on deep neural network by using IGASAA, and alert system: This module creates alert to the admin about intrusion detected by the detection module. They placed their proposed model in both inside and outside of the cloud environment on CloudSim 4.0.

A reinforcement learning model for IDS called adversarial environment reinforcement learning (AE-RL) was discussed by Caminero *et al.* (2019) [26]. NSL-KDD and AWID data set were used for testing the algorithms. They perform the comparison among machine learning, deep learning, and deep reinforcement learning algorithm such as SVM, with kernel and radial basis function kernel, RF, MLP, gradient boosting machine (GBM), CNN, and deep reinforcement learning (DLR). Their result observation shows that AE-RL provides an F1 Score closely to SVM-RBF, but AE-RL has the highest detection rate even with fewer labels and requires less training time and prediction time. The author proved that AE-RL is suitable for unbalanced data. Martin *et al.* (2020) [14] proposed a randomized ensembles-based deep learning architecture for the early identification of Alzheimer's disease and overcame the problem of overfitting. Unlike conventional machine learning algorithms, deep learning can handle this variance in attributes and samples.

Because most attacks use IP/port address information, Martin *et al.* (2021) [5] presented a feature that can anticipate the co-occurrence of source and destination. The original network address is replaced based on the stated distance between various components of source and destination (IP and port addresses). A neural network with hash functions is used to incorporate a network of distinct network addresses. By doing this address replacement, the prediction of network intrusions has been improved, and the author also tested the improvement of the proposed address replacement with the CICDS2017 and CICDS2019 intrusion dataset.

Zhang *et al.* (2020) [27] designed a gradient-free approach to show that RF is more vulnerable to cyberattacks than SVM. They showed how hostile inputs modified only on the model decision outputs can easily elude a discrete-valued random forest classifier.

To predict exploitation time of vulnerability assessment, Tang *et al.* (2021) [28] proposed an adaptive sliding window weighted learning that outfits the problem of dynamic imbalanced multiclass that usually appears in all industries.

Huang *et al.* (2021) [29] introduced a multi-scale guided feature extraction and classification (MGFEC) algorithm for extracting the features from hyperspectral images. They proved that MGFEC outperforms than random patch network algorithm (RPNet). The dataset characteristics determine the accuracy and classification of any model's performance (Chen *et al.*, 2020) [30]. If the data collection contains many variables and features, it is critical to concentrate on feature selection. In some cases, such as the medical and security domains, feature selection is quite tough. Iman *et al.* (2020) [21] used the Bortua feature selection algorithm [31] to extract the important features from the dataset, and hence they proved that the classifier's performance has improved.

This research article proposed Boruta feature selection with grid search random forest (BFS-GSRF) algorithm to enhance the classification algorithm's accuracy and focus on the feature selection method.

III. Dataset Description

Implementing the proposed model for detecting intrusion has been done on the knowledge on discovery CUP (KD-

DCUP) dataset [8]. The dataset descriptions are given in Table I [32].

KDD CUP data set consists of nearly 490,000 observations with 42 features [8]. For intrusion detection, the majority of the researchers (Saranya *et al.*, 2020 [8]; Yaseen *et al.*, 2017 [15]) used the KDDCUP dataset in their work. KDD-CUP is the larger dataset when compared to NSL-KDD and UNSW-NB15. The label feature has two major categories normal and attack. The attack has 24 types which fall under four categories such as denial of service (DoS), remote to local attack (R2L), user to remote (U2R), and probing. The attack details are given in Table II.

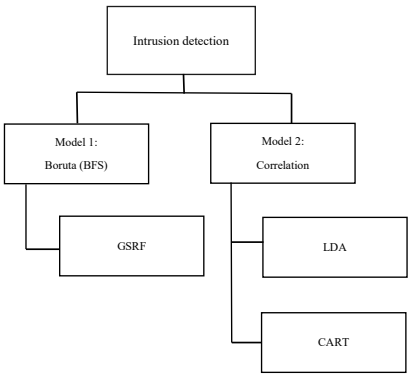

Fig. 1. Workflow of the proposed model.

IV. Proposed Model for Intrusion Detection System (IDS)

The paper mainly focuses on effective feature selection to attain higher accuracy on IDS. This paper has proposed two models: Model 1: Features are selected using the wrapper approach, and random forest is used for classification. Model 2: Two filter methods are used for feature selection and linear discriminant analysis; the CART algorithm is used for classification. The model of the proposed algorithm is shown in Fig. 1.

The steps involved in the proposed model include pre-processing, feature selection, and classification. In pre-processing, the data set structure is viewed, the attribute's data type is changed as per the algorithm needs, and checks for missing values. Then, the label of pre-processed data is mapped with the appropriate class of attacks. The pre-processing step is common to both model 1 and model 2.

In model 1, features are selected using the wrapper approach. Based on the classifier performance, the wrapper-based method measures the usability of features. The data set used in this research is specifically collected for network traffic. It is not reliant on assumptions. Only feature extraction through the Boruta algorithm was embedded in the proposed work. Boruta is a useful algorithm for feature selection when there are many features. No assumptions are made in the dataset.

Boruta algorithm is one of the wrapper-based algorithms used for feature selection. For intrusion detection, the KDD-CUP dataset is used in the proposed work. The dataset has 42 features, but not every feature is important for predicting attacks or intrusions. This proposed work aims to have an efficient feature selection to achieve higher accuracy for intrusion detection. The Boruta algorithm is used to eradicate redundant variables and identify important variables, which returns a precise classification and robust model.

The proposed Boruta feature selection with grid search random forest (BFS-GSRF) algorithm is used for feature selection and classification. The wrapper-based feature selection method uses a prediction algorithm to select a subset of features. Each subset trains a new model and provides a better-performing feature set that will yield a reasonable accuracy. The Boruta algorithm works well for big data and the data set, which has more features. It is used to select the most significant and interesting features in the data set. Variable selection is a vital part of building an intrusion detection system, which will produce a model free from noise and false predictions.

The proposed BFS-GSRF is based on the idea from the random forest classifier [30] by summing the randomness to the model and gathering results from the ensemble of the randomized set. The BFS-GSRF usually ran without tuning the parameters and provides a numerical approximation of the important feature importance. The Z-score is not directly linked to the statistical significance of the feature set that comes from the random forest algorithm. This Z-score will need the external reference to fix the influenced attributes. So, this paper used Z-score as the significant measure in Boruta feature selection with random forest (BFS-GSRF). The steps and workflow of BFS are shown in Figs. 2 and 3.

For classification, random forest with grid search (RFGS) is used. RF is one of the widely used algorithms for classification problems. This algorithm will create several classification trees for predicting the target class. Based on the majority of the vote, the final prediction was made. Parameter optimization is used to improve the accuracy of the RF algorithm. The grid search method is used in RF to obtain the classification model with higher accuracy for tuning the parameter. The randomly based search method is more efficient than the grid-based search method for hyperparameter optimization. Two discrete integer parameters, such as ntree and mtry, are used to tune the parameter. The main objective of the optimization is to minimize the out of bag (OOB) error. After multiple runs, optimal parameters value is chosen based on the pair that produces the lowest OOB error. Based on that parametric value selected, the tree was built. The workflow of BFS-GSRF is shown in Fig. 4.

The features in model 2 are chosen based on Pearson's correlation coefficient. The correlation-based feature selection is supported using the ranker search method using the correlation attribute evaluation technique. The linear discriminant analysis (LDA) and classification and regression tree (CART) algorithm are used. LDA is a supervised linear machine learning technique that is commonly used to reduce dimensionality and classify data. LDA creates class separation by sketching a decision area between the different classes present in the dataset. LDA works well for multiclass classification problems. LDA provides maximal separation by increasing the

TABLE I
FEATURE DESCRIPTION OF KDDCUP DATASET.

No	Features name	Descriptions	Type
1	duration	Connection time	Continuous
2	protocol_type	Protocol type	Symbolic
3	service	Destination network service	Symbolic
4	src_bytes	No. of bytes from source to destination	Continuous
5	dst_bytes	No. of bytes from destination to source	Continuous
6	Flag	Status of the connection	Symbolic
7	Land	1 = connection from same host/port, else 0	Symbolic
8	wrong_fragment	No. of wrong fragments	Continuous
9	Urgent	No. of urgent packets	Continuous
10	Hot	No. of hot indicators	Continuous
11	num_failed_login	No. of failed logins	Continuous
12	logged_in	1 = successfully logged in, else 0	Symbolic
13	num_compromised	No. of compromised	Continuous
14	root_shell	1 = root shell, else 0	Continuous
15	su_attempted	1 = su root command, else 0	Continuous
16	num_root	No. of root access	Continuous
17	num_file_creations	No. of operations on file creation	Continuous
18	num_shells	No. of shell prompts	Continuous
19	num_access_files	No. of access control files operations	Continuous
20	num_outbound_cmds	No. of outbound commands on ftp session	Continuous
21	is_hot_login	1 = hot login list, else 0	Symbolic
22	is_guest_login	1 = guest login, else 0	Symbolic
23	Count	Same no. of host connection as the current connection in past two seconds	Continuous
24	serror_rate	Percentage of SYN error connection	Continuous
25	rerror_rate	Percentage of REJ error connection	Continuous
26	same_srv_rate	Percentage of same service connections	Continuous
27	diff_srv_rate	Percentage of different service connections	Continuous
28	srv_count	Same no. of service as the current connection in past two seconds	Continuous
29	srv_serror_rate	Percentage of connection with SYN errors	Continuous
30	srv_rerror_rate	Percentage of connection with REJ errors	Continuous
31	srv_diff_host_rate	Percentage of different host connection	Continuous
32	dst_host_count	Count of same destination host connection	Continuous
33	dst_host_srv_count	Count of same destination host connection using same service	Continuous
34	dst_host_same_srv_rate	Percentage of same destination port connection using same service	Continuous
35	dst_host_diff_srv_rate	Percentage of current host on different service	Continuous
36	dst_host_same_src_port_rate	Percentage of same source port on current host	Continuous
37	dst_host_srv_diff_host_rate	Percentage of same service connection from different host	Continuous
38	dst_host_serror_rate	Percentage of current host connection having s0 error	Continuous
39	dst_host_srv_serror_rate	Percentage of current host and specified service connection having s0 error	Continuous
40	dst_host_rerror_rate	Percentage of current host connection with RST error	Continuous
41	dst_host_srv_rerror_rate	Percentage of current host connection and specified service with RST error	Continuous
42	connection_type	Normal or attack	Continuous

TABLE II
ATTACK TYPES.

Major attack	Explanation	Attack types
DoS	This attack makes the resource too busy or even unavailable to the legitimate users	Back, Land, Pod, smurf, Neptune, teardrop
R2L	The attacker sends a packet to the local system remotely without having an account on that machine by exploiting vulnerabilities	Guess_Password, Imap, Phf, Warezclient, Warezmaster, Spy, Multihop, Ftp_write
U2R	The attacker illegally attained root access to the machines by exploiting some vulnerabilities on the target machine	Buffer_overflow, Rootkit, perl, Loadmodule,
Probe	To determine the vulnerabilities, the attacker scans the network or a system	Satan, Ipsweep, Nmap, Portsweep

ratio of between-class variation to within-class variance. These algorithms use the Bayes theorem to calculate the likelihood that incoming inputs belong to which class.

$$P\left(Y = x \mid X = x\right) = \frac{(P|k * fk\left(x\right))}{sum(P|| * f\left(x\right))}, \tag{1}$$

where the output class is $k(x)$, the input class is x, the estimated probability is $f(x)$, and the prior probability is $P|k$. When using LDA to solve a classification problem, the output variable should be categorical and support binary and multiclass classification.

CART classification is a supervised nonlinear algorithm used for classification and regression. This algorithm constructs the binary decision tree by splitting the attributes, which is considered as a node. The whole tree from root to leaf contains a learning sample. For classification, the target variable in CART should be categorical, whereas the target variable for the regression tree should be continuous. Here the target variable is categorical for performing classification on intrusion detection. The metric Gini index is used to perform the classification task. The Gini index will store the squared probabilities of each class.

$$Giniindex\ = 1 - \sum_{i=1}^{c} (P_i)^2, \tag{2}$$

1. The algorithm creates Shadow features by duplicate copies of the dataset, and the values are shuffled in all columns
2. The original data set values are combines with shadow values.
3. After combining, random forest classifier is used on thecombined dataset so that the variable importance is measured, by default it uses mean decrease accuracy.
4. The Boruta checks for higher importance of original features bycomputing Z-score and finding the maximum Z-score among the shadow attributes.
5. The Z-score of original values and shuffled values are compared atevery iteration to see the better one than the existing one
6. To increase the robustness, the boruta validate the importance of the feature by comparing with random shuffled copies.
7. Higher the score is considered to be higher importance.

Fig. 2. Steps of Boruta algorithm.

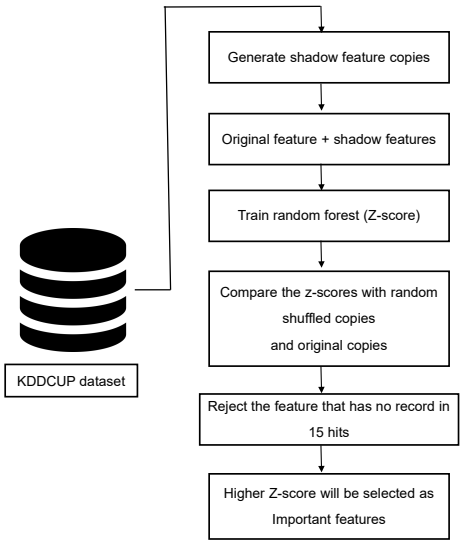

Fig. 3. Workflow of Boruta feature selection.

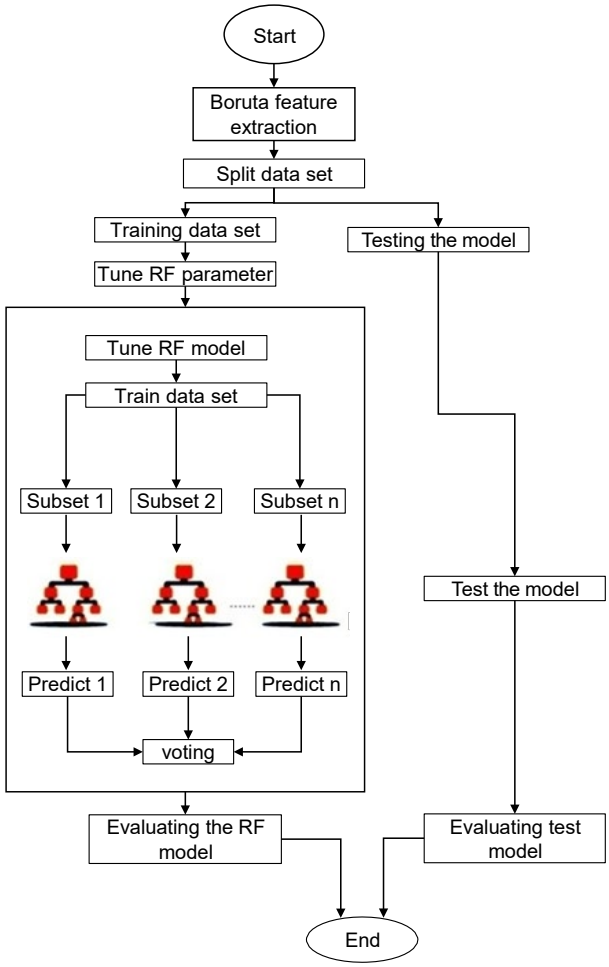

Fig. 4. Process flow of proposed model – BFS-GSRF algorithm.

where c is the number of classes, and the probability of each class in the dataset is Pi. Accuracy and kappa are the metrics used to measure the classification performance of the proposed model. Accuracy shows how the model is close to the truth. It is the percentage of exact classification out of all instances. It can be calculated by the formula:

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

where TP is the true positive, TN is the true negative, FP is the false positive, and FN is the false negative. Kappa or Cohen's kappa is similar to accuracy, and it is used to measure inter-rater reliability items. It can be calculated by the formula:

$$K = \frac{p_o - p_e}{1 - p_e}, \tag{4}$$

where p_o is the observed agreement, and p_e is the hypothetical probability.

V. Results and Discussion

The results of both model 1 and model 2 for intrusion detection are shown in this section. The proposed model was tested on the standard KDDCUP data set in R studio. The tests were carried out on a personal PC that was running Windows 10. This research aims to improve intrusion detection by using intelligent feature selection and better categorization. In model 1, Boruta feature selection (BFS) is used for feature selection, and grid search random forest (GSRF) algorithm is used for classification. In model 2, correlations-based feature selection and LDA and CART are used for classification. When comparing model 1 with model 2 and literature work, model 1 achieves better accuracy.

BFS selects 26 features out of 42 features. The selected features are used for classification with GSRF. The mean decrease gini is used to measure the variable importance of the target features. The importance of the target variable is shown in the form of quantitative value in Fig. 5. The parameter tuning with grid search is proposed in this research. For parameter tuning, the number of tree size (n_{tree}) is also significant; as the number of trees grows, the performance of the forest is also significantly high, but there is no significant gain when the tree size get double or beyond some threshold and even if the tree size is very small the forest will not yield better performance. The tree size 200 (n_{tree} = 200) produces better performance for intrusion detection.

To measure the classifier's performance, in the existing literature (Caminero *et al.*, 2019) and (Saranya *et al.*, 2020),

	MeanDecreaseGini
srv_rerror_rate	37.66397
rerror_rate	41.16974
flag	382.89516
dst_host_rerror_rate	102.73237
logged_in	1711.06349
dst_bytes	2778.69340
src_bytes	944.37672
num_compromised	118.42172
dst_host_srv_count	283.47827
duration	64.28718
dst_host_same_src_port_rate	398.95285
dst_host_diff_srv_rate	403.85706
dst_host_count	1298.83724
dst_host_srv_serror_rate	60.76001
count	4026.39501
hot	126.09113
dst_host_same_srv_rate	255.54603
dst_host_srv_diff_host_rate	833.95356
dst_host_serror_rate	115.58663
serror_rate	83.24600
srv_serror_rate	34.77921
diff_srv_rate	656.27936
srv_count	777.72003
srv_diff_host_rate	247.49379
protocol_type	665.41816

Fig. 5. Variable selection through Boruta algorithm.

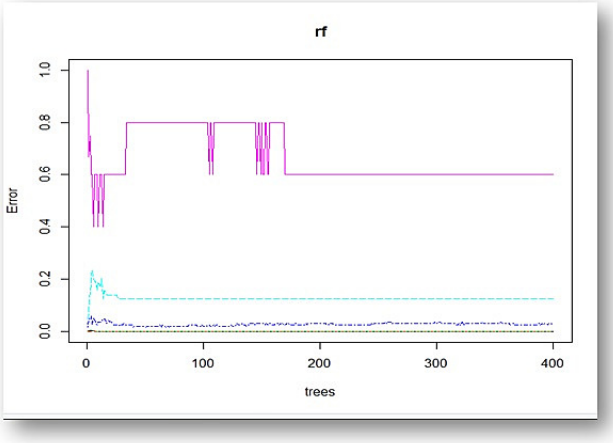

Fig. 6. The error rate of GSRF.

they used the metrics like accuracy and error rate. In the medical diagnosis and security applications, they are mainly focusing on false positive and false negative rates rather than focusing on accuracy. Hence in the proposed work, the author measured the performance of the classifiers using metrics like sensitivity, specificity and accuracy, which is shown in Tables IV–VI. To measure the inter-rater reliability items, the metics' Kappa is used. Kappa value of LDA and CART algorithm is shown in Table VI. The prediction error of random forest can be measured by the OOB error method. As the number of trees grows, this graph shows that the OOB error rate initially falls and becomes more constant after 200 trees. The OOB error is high (0.09) at the m_{try} of 2, and then it

TABLE III
ACCURACY OF THE PROPOSED MODEL.

Results across tuning RF parameters		
m_{try}	Accuracy	Kappa
2	0.9963	0.9890
18	0.9991	0.9975
35	0.9987	0.9962

TABLE IV
CLASSIFICATION USING THE PROPOSED MODEL.

Method 1: Confusion matrix of IDS (BFS-GSBRF)					
Prediction	DoS	Normal	Probe	R2L	U2R
DoS	352262	31	24	2	0
Normal	49	87485	42	73	30
Probe	1	15	3630	3	0
R2L	0	18	0	935	4
U2R	0	1	0	0	12
Statistics of method 1					
Class	DoS	Normal	Probe	R2L	U2R
Sensitivity	0.9999	0.9997	0.9817	0.9180	0.9012
Specificity	0.9997	0.9995	0.9999	0.9999	0.9904

comes down when the m_{try} of 18. On the contrary, when m_{try} is equal to 35 and beyond 35, the OOB error increases again. The error rate for the proposed random forest is shown in Fig. 6.

The accuracy and kappa of the proposed model for the m_{try} of 2, 18, and 35 is shown in Table III. The prediction of the proposed model is given in Table IV. This Table shows that the proposed work achieves good prediction and the accuracy of the proposed BFS-GSRF is 99.9% for the m_{try} of 18.

The evaluation of model 2 is similar to that of model 1. The accuracy and kappa of LDA and CART is shown in Table IV to VI. The LDA achieves an accuracy of 98.3% and CART achieves an accuracy of 98%. When comparing LDA and CART, LDA works better than CART because CART predicts only DoS attacks.

Receiver operating characteristics (ROC) curve is the plot that shows a trade-off between sensitivity and specificity. In ROC, all points that reside on the upper-diagonal region are corresponding to good classifiers. All points, which reside in the lower-diagonal region, are corresponding to worse classifiers. All points, which reside in the upper-diagonal region, have lower FPR than TPR. The ROC curve of the proposed model is shown in Fig. 7. The proposed model BFS-GSRF yields better accuracy since all points reside on the upper-diagonal region.

Fig. 8 shows the comparisons of the proposed model. This chart indicates that the proposed model BFS-GSRF is the best classifier than LDA and CART. However, the correlation-based feature selection is fast scalable and better than wrapper-based Boruta feature selection in computational complexity. Still, it ignores the interaction with the classifier; this makes the LDA and CART classifier less accurate. The BFS yields robust results for feature selection in IDS [29] than the usual correlation method. The BFS is achieved by running a random forest classifier on both original and random features to compute the importance of variables. In wrapper-based BFS feature, dependencies are modeled and interact with the classifier and interact with variables, so it is less prone to

TABLE V
CLASSIFICATION USING LDA.

Method 2: Confusion matrix of IDS (LDA)					
Prediction	DoS	Normal	Probe	R2L	U2R
DoS	77631	126	18	0	0
Normal	537	18688	59	15	3
Probe	47	114	739	1	0
R2L	0	36	0	66	0
U2R	76	491	5	143	7
Statistics of method 1					
Class	DoS	Normal	Probe	R2L	U2R
Sensitivity	0.9919	0.9606	0.9001	0.8202	0.7909
Specificity	0.9930	0.9923	0.9983	0.9996	0.8962

TABLE VI
ACCURACY, KAPPA OF LDA AND CART.

Accuracy of model: 2		
Model	Accuracy	Kappa
LDA	0.9831	0.9499
CART	0.9803	0.9406

Fig. 7. ROC curve of the proposed model BFS-GSRF.

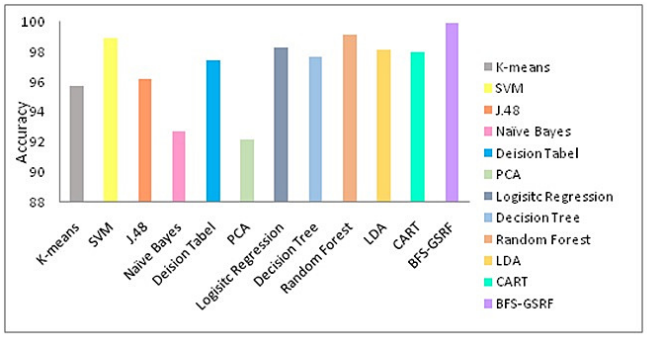

Fig. 8. Comparisons of the proposed model with other machine learning algorithms.

local optima. The whole process of BFS is dependent on permuted copies, and the random permutation process is repeated until the statistically robust result is obtained. For checking all parameter combinations, this paper proposed a hyper-parameterized grid search method (GFRS) on random forest, and this exhaustive method helps find optimal hyper-parameter values. The optimal hyper-parameter value is obtained from BFS-GSRF and yields a robust outcome with 99.9% accuracy.

When there are a large number of features, Boruta is a useful algorithm for feature selection. However, Boruta is prone to an infinite loop, which can be avoided by combining the Gini index with random forest. Because the work was accomplished using R programming, this state of the art is good enough to provide improved accuracy with superior multi-classification, but it is limited in terms of time. These limits can be overcome by utilizing higher-level platforms.

VI. CONCLUSIONS AND FUTURE WORK

This research paper proposed a novel BFS-GSRF for network intrusion detection systems. The model is evaluated on well-known standard datasets, KDDCUP. The performance of the classifiers such as SVM, LDA, CART and the random forest is 98.5%, 98%, 97.7%, and 99%, respectively. To improve the performance of the classifier further, BSF-RF algorithm is introduced in the proposed work. The BFS-RF is used for efficient feature selection based on wrapper and ensemble techniques. BFS-RF performance is assessed in terms of accuracy and achieved 99.9%. This work is also compared with LDA and CART machine learning algorithms. In the future, the work will focus on reducing the training time and building an efficient classifier that classifies upcoming new attacks by speeding up the data analysis performance and deploying it in a real-time environment.

REFERENCES

[1] P. Sinha, V. K. Jha, A. K. Rai, and B. Bhushan, "Security vulnerabilities, attacks and countermeasures in wireless sensor networks at various layers of osi reference model: A survey," in *Proc. IEEE ICSPC*, 2017
[2] Y. Maleh, A. Ezzati, Y. Qasmaoui, and M. Mbida, "A global hybrid intrusion detection system for wireless sensor networks," *Procedia Comput. Sci.*, vol. 52, pp. 1047–1052, 2015.
[3] E. Kabir, J. Hu, H. Wang, and G. Zhuo, "A novel statistical technique for intrusion detection systems," *Future Gener. Comput. Syst.*, vol. 79, pp. 303–318, 2018.
[4] L. Fernandez Maimo *et al.*, "A self-adaptive deep learning-based system for anomaly detection in 5G networks," *IEEE Access*, vol. 6, pp. 7700–7712, 2018.
[5] M. Lopez-Martin, B. Carro, J. I. Arribas, and A. Sanchez-Esguevillas, "Network intrusion detection with a novel hierarchy of distances between embeddings of hash IP addresses," *Knowledge-based Syst.*, vol. 219, 2021.
[6] Z. Chen, F. Han, L. Wu, J. Yu, S. Cheng, P. Lin, and H. Chen, "Random forest based intelligent fault diagnosis for PV arrays using array voltage and string currents," *Energy Convers. Manage.*, vol. 178, pp. 250–264, 2018.

[7] A. B. Abhale and S. S. Manivannan, "Supervised machine learning classification algorithmic approach for finding anomaly type of intrusion detection in wireless sensor network," *Opt. Memory Neural Netw.*, vol. 29, no. 3, pp. 244–256, 2020.
[8] T. Saranya, S. Sridevi, C. Deisy, T. D. Chung, and M. K. A. A. Khan, "Performance analysis of machine learning algorithms in intrusion detection system: A review," in *Procedia Comput. Sci.*, vol. 171, pp. 1251–1260, 2020.
[9] H. Wang, J. Gu, and S. Wang, "An effective intrusion detection framework based on SVM with feature augmentation," *Knowledge-based Syst.*, vol. 136, pp. 130–139, 2017.
[10] U. Abirami and S. Sridevi, "Traffic flow prophecy with mapreduce job for big data driven," in *Proc. IEEE ICoAC*, 2017.
[11] M. C. Belavagi and B. Muniyal, "Performance evaluation of supervised machine learning algorithms for intrusion detection," in *Procedia Comput. Sci.*, vol. 89, pp. 117–123, 2016.
[12] L. Li, H. Zhang, H. Peng, and Y. Yang, "Nearest neighbors based density peaks approach to intrusion detection," *Chaos, Solitons Fractals*, vol. 110, pp. 33–40, 2018.
[13] S. Sridevi, S. Parthasarathy, and S. Rajaram, "An effective prediction system for time series data using pattern matching algorithms," *Int. J. Ind. Eng.: Theory Appl. Pract.*, vol. 25, no. 2, pp. 123–136, 2018.
[14] M. Lopez-Martin, A. Nevado, and B. Carro, "Detection of early stages of alzheimer's disease based on meg activity with a randomized convolutional neural network," *Artif. Intell. Medicine*, vol. 107, 2020.
[15] W. L. Al-Yaseen, Z. A. Othman, and M. Z. A. Nazri, "Multi-level hybrid support vector machine and extreme learning machine based on modified k-means for intrusion detection system," *Expert Syst. Appl.*, vol. 67, pp. 296–303, 2017.
[16] S. Masarat, S. Sharifian, and H. Taheri, "Modified parallel random forest for intrusion detection systems," *J. Supercomputing*, vol. 72, no. 6, pp. 2235–2258, 2016.
[17] S. U. Jan, S. Ahmed, V. Shakhov, and I. Koo, "Toward a lightweight intrusion detection system for the Internet of things," *IEEE Access*, vol. 7, pp. 42 450–42 471, 2019.
[18] R. Vijayanand, D. Devaraj, and B. Kannapiran, "Intrusion detection system for wireless mesh network using multiple support vector machine classifiers with genetic-algorithm-based feature selection," *Comput. Secur.*, vol. 77, pp. 304–314, 2018.
[19] O. Y. Al-Jarrah, Y. Al-Hammdi, P. D. Yoo, S. Muhaidat, and M. Al-Qutayri, "Semi-supervised multi-layered clustering model for intrusion detection," *Digit. Commun. Netw.*, vol. 4, no. 4, pp. 277–286, 2018.
[20] S. M. Othman, F. M. Ba-Alwi, N. T. Alsohybe, and A. Y. Al-Hashida, "Intrusion detection model using machine learning algorithm on big data environment," *J. Big Data*, vol. 5, no. 1, 2018.
[21] A. N. Iman and T. Ahmad, "Improving intrusion detection system by estimating parameters of random forest in Boruta," in *Proc. IEEE ICoSTA*, 2020.
[22] Z. Chiba, N. Abghour, K. Moussaid, A. El omri, and M. Rida, "Intelligent approach to build a deep neural network based IDS for cloud environment using combination of machine learning algorithms," *Comput. Secur.*, vol. 86, pp. 291–317, 2019.
[23] M. Hasan, M. M. Islam, M. I. I. Zarif, and M. M. A. Hashem, "Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches," *Internet Things*, vol. 7, 2019.
[24] S. Dey, Q. Ye, and S. Sampalli, "A machine learning based intrusion detection scheme for data fusion in mobile clouds involving heterogeneous client networks," *Inf. Fusion*, vol. 49, pp. 205–215, 2019.
[25] X. An, J. Su, X. Lü, and F. Lin, "Hypergraph clustering model-based association analysis of DDoS attacks in fog computing intrusion detection system," *Eurasip J. Wireless Commun. Netw.*, vol. 2018, no. 1, 2018.
[26] G. Caminero, M. Lopez-Martin, and B. Carro, "Adversarial environment reinforcement learning algorithm for intrusion detection," *Comput. Netw.*, vol. 159, pp. 96–109, 2019.
[27] F. Zhang, Y. Wang, S. Liu, and H. Wang, "Decision-based evasion attacks on tree ensemble classifiers," *World Wide Web*, vol. 23, no. 5, pp. 2957–2977, 2020.
[28] T. M. C. J. W. H. Y. M. Yin, J. and Y. Lin, "Vulnerability exploitation time prediction: an integrated framework for dynamic imbalanced learning," *World Wide Web*, pp. 1–23, 2021.
[29] S. Huang, Y. Lu, W. Wang, and K. Sun, "Multi-scale guided feature extraction and classification algorithm for hyperspectral images," *Scientific Reports*, vol. 11, no. 1, 2021.
[30] R. . Chen, C. Dewi, S. . Huang, and R. E. Caraka, "Selecting critical features for data classification based on machine learning methods," *J. Big Data*, vol. 7, no. 1, 2020.
[31] R. Tang and X. Zhang, "Cart decision tree combined with Boruta feature selection for medical data classification," in *Proc. IEEE ICBDA*, 2020.
[32] "KDD CUP 1999 data," [Online] Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

Sridevi Subbiah is working as an Associate Professor in the Information Technology Department, Thiagarajar College of Engineering, Madurai, Tamilnadu, India since 2006. She has the self-drive and motivation for publishing her research work in reputed journals and conferences. She published more than 40 articles in journals and conferences. She acts as a Reviewer in various SCI and Scopus indexed technical journals. She is an active member of ACM and CSI. Her research area interest includes temporal data analytics, computer graphics, data science and machine learning.

Kalaiarasi Sonai Muthu Anbananthen is an Associate Professor in the Faculty of Information Science and Technology at Multimedia University (MMU), Malaysia. She was a Programme Coordinator for the Masters of Information Technology (Information System). She acts as a Reviewer in various Scopus and SCI indexed technical journals. She has published more than 80 articles in journals, conferences and book chapters. Her current research interests focus on data mining, sentiment analysis, artificial intelligence, machine learning, deep learning and text analytics.

Saranya Thangaraj is a Research Scholar doing. Her research includes information security, deep learning, cyber-physical systems and IoT

Subarmaniam Kannan has been a Lecturer in the Faculty of Information Science and Technology, Multimedia University, since 2000. He has a Ph.D. in Semantic Learning (Knowledge Engineering) from Multimedia University. He is also a Certified Information Systems Auditor (CISA) and Certified Cisco Networking Associate (CCNA) Registrar and Instructor for MMU-Melaka Local Networking Academy. He was Programme Coordinator for Data Communications and Networking Programme from 2013 to 2021. His research area includes semantic web technology, ontology and knowledge management, automatic speech recognition for Bahasa Malaysia and edge computing analytics.

Deisy Chelliah is working as a Professor and Head of the Information Technology Department, Thiagarajar College of Engineering, Madurai, Tamilnadu, India. She published more than 70 articles in journals and conferences. She acts as a Reviewer in various SCI and Scopus indexed technical journals. She completed two projects sponsored by Microsoft and AICTE. She is a Member of ISTE and CSI. Her research area interest includes image analysis and text analytics.