International Journal of Safety and Security Engineering
Vol. 15, No. 11, November, 2025, pp. 2219-2227
Journal homepage: http://iieta.org/journals/ijsse

Design and Evaluation of BE-RF Framework on Multi-Dataset: A Network Intrusion Detection System Using Ensemble Learning with Random Forest

Mukhtar Ghaleb

Department of Information Systems and Cyber Security, College of Computing and Information Technology, University of Bisha, Bisha 61922, Saudi Arabia

Corresponding Author Email: mghaleb@ub.edu.sa

https://doi.org/10.18280/ijsse.151103

Received: 8 September 2025
Revised: 9 November 2025
Accepted: 20 November 2025
Available online: 30 November 2025

Keywords:
cyber-attack, intrusion detection, machine learning, intelligent security, random forest, ensemble learning

ABSTRACT

Cyber-attacks and associated challenges have caused a relentless loss of money, data, and a tragic impact on the personal and public levels. These attacks did not spare even the most critical infrastructures of the smart grid and nuclear facilities. Consequently, this has reinforced the general trend towards searching for appropriate means and techniques to prevent, reduce, or mitigate the risk of cyber-attacks. Recently, the use of artificial intelligence in various fields has proven its effectiveness due to its ability to devise fast learning and highly accurate learning models. Therefore, in this paper, we used machine-learning techniques to learn the patterns of cyberattacks and build an accurate classification model of data flow in networks to take advantage of the intelligent machine's capabilities to identify potential attacks. We propose using the Bagging Ensemble Learning method fortified by the Random Forest (BE-RF) to build a classification model for the attack categories. The random forest algorithm generates many decision trees and then combines them to obtain the most accurate threat classifier, while the bagging ensemble improves the stability and accuracy of machine learning (ML) algorithms. We employ UNSW-NB15, NSL-KDD, and CICIDS2017 datasets for evaluating the classification performance over several previously used classifiers, such as AlexNet, Convolutional Neural Network (CNN), and Bidirectional Long Short-Term Memory network (BiLSTM). According to the F1 assessment, the proposed BE-RF model achieved results of 90.43 on the UNSW-NB15 dataset, 84.9 on the NSL-KDD dataset, and 99 on the CICIDS2017 dataset. The results show the success of the proposed methodology in terms of accuracy, precision, recall, and F1 Measure compared to previous methods.

1. INTRODUCTION

Advances in internet technology have become a daily necessity for everyone, radically changing their lives. However, the internet remains vulnerable to an increasing number of attacks, making network security a critical issue. This has led researchers to emphasize the need to use the appropriate technologies, such as AI, to overcome and anticipate attacks before they occur as a preventative measure. To give the reader a clear understanding, the following section will describe the Intrusion Detection System (IDS) and then highlight the importance of applying machine learning (ML) to them.

1.1 Intrusion Detection System

One of the most serious problems in network security is intrusions, which can also cause damage to system hardware. An IDS is a piece of hardware or software that monitors malicious activity or policy violations on a network or other systems. IDS has become an important field in the world of information security; it is to find the assaults and take a set of precautions to disable them and prevent the losses arising from these attacks [1]. There are many different IDS in use, but accuracy is one of the main issues. The detection rate and false alarm rate play a major role in the accuracy analysis. Researchers work on improving IDS to minimize false alarms and increase detection rates. Applications for IDS types range widely, from small networks to many machines. Two famous types are Host-based IDS (HIDS) and Network Intrusion Detection Systems (NIDS) [2]. IDS detects intrusions or attacks by evaluating audit data and raising an alarm. HIDS is installed on a single system, known as the target system, that is thought to be vulnerable to assault. HIDS tries to detect any assault by evaluating changes in the system log files. It is possible to bypass HIDS in case of any malfunction in the target device's operating system since HIDS is installed on the target device [3].

Several studies proposed different techniques to improve the IDS performance. The study uses multistage deep learning image recognition to suggest a unique method for network intrusion detection [4]. The work by Wang et al. [5] provides an IDS based on a combination of prediction and learning processes to increase the accuracy of anomaly detection. The

prediction model is based on the Kalman filter, while the learning method is based on automated ML. The adaptability performance of the model is to be enhanced by combining group convolution with a unique IDS approach as introduced by Imran et al. [6]. To improve the detection performance, Jaw and Wang [7] proposed a hybrid feature selection (HFS) with an ensemble classifier, which chooses pertinent features and offers reliable attack categorization, has been presented. In the energy sector, not all companies are using IDS to protect their process control networks (PCNs). However, it is believed that IDS systems can provide proper protection for these interconnected smart energy systems, including the smart grid [8].

1.2 Machine learning

Due to its ability to learn and make decisions, ML has become a fundamental tool for utilizing AI technology. To help computers evolve and improve continuously, ML automatically creates an intelligent model using training data without the need for traditional training methods. The two most frequently used ML approaches are supervised and unsupervised learning. This has helped make ML an integral part of modern business and research across all sectors. The use of ML-led techniques in anomaly-based detection has risen in prominence in recent years. Consequently, combining ML with information security led to the rise of intelligent security to automate proactive threat detection, as illustrated by Figure 1.

Cyber-attacks have ravaged/invaded almost every computer network connected to the internet, including the critical electrical smart power grid, renewable energy systems, and even nuclear power plants. Initially, many related studies discussed the problems of detecting and identifying attacks without using machine-learning techniques [9-11]. However, combining AI and information security has led to what is known as Intelligent Security. The aim of intelligent security is to benefit from AI to build a model that can detect any possible threat. IDS are critical for network security because they send out alarms whenever they detect aberrant network traffic [12]. The use of ML-based anomaly detection techniques, mainly IDS and IPS, can greatly help protect these critical infrastructures against cyber-attacks, including the infamous zero-day exploits [13, 14].

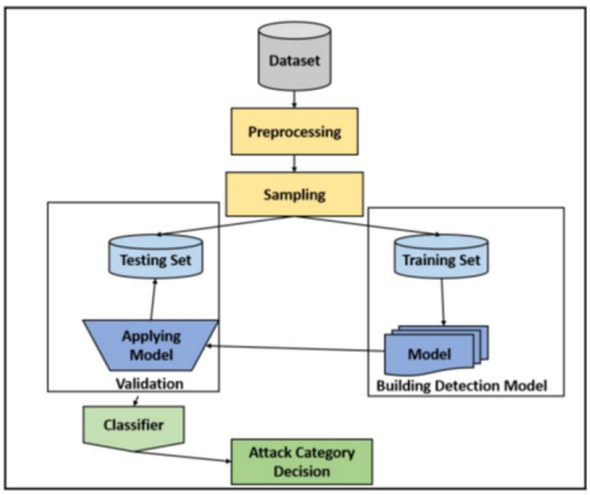

Figure 1. Intelligent security architecture

For instance, as this research suggests, combining different models improves ML results for building a cyberattack classifier. One such successful methodology is the use of Ensemble ML algorithms, which have recently proven their ability to make more accurate predictions [15]. Therefore, we propose to benefit from combining several methods to gain the best accuracy for IDS in terms of precision, recall, and F1 measures compared to previous methods such as AlexNet, Convolutional Neural Network (CNN), and Bidirectional Long Short-Term Memory network (BiLSTM). This includes handling unbalanced data, proposing the combination of Ensemble learning with Random Forest. Following that is testing the proposed methodology, which we named Random Forest (BE-RF), on three datasets, namely, UNSW-NB15, NSL-KDD, and real-time CICIDS2017.

2. RELATED WORKS

Researchers found it useful to use an ML algorithm for IDS. They tested their proposed methods on numerous documented datasets to ensure reliability and transparency. For instance, an important study proposed using ML-based detection strategies for new attacks using variants in the UNSB-NB15 as a large dataset [16]. To achieve high accuracy in the identification of attacks in the respected dataset, they employ ML classifiers. A high level of accuracy of 86.15 percent was attained. Another dataset, namely NSL-KDD, was also used in a comparative study along with K-Nearest Neighbors (KNN), Naive Bayes, Support Vector Machines (SVM), and Decision Tree for testing and training of the proposed Network Intrusion Detection Model [17]. When tested independently, with the exception of Naive Bayes, three of the four ML algorithms showed significant performance improvement for detecting dangerous network intrusions.

Researchers then turned their attention to using a very promising branch of ML: neural networks (NN). Chen et al. [18] suggested a convolutional Neural Network-based network IDS (NN-IDS). Their findings confirm the system's efficiency and demonstrate that the accuracy of the detection engine trained using raw traffic is better than that of the detection engine trained using obtained characteristics. Yang et al. [19] present a hybrid IDS that combines MDPCA with deep belief networks (DBNs). MDPCA is a technique for identifying common features in complicated and wide-ranging network data. The suggested model outperforms other well-known classification approaches in terms of overall accuracy, recall, precision, and F1-score on the NSL-KDD and UNSW-NB15 datasets [20]. Extreme Learning Machine and Artificial Immune System (AIS-ELM) are used by Alalade [21] to give early work on an ID that uses them to detect anomalies in a smart home network. Another study tested and compared various ML-based anomaly detection methods that can be used for intrusion detection in electrical substations of smart grids [22]. They used network traffic test data from the Austrian power grid to study four different attacks against the grid. Li et al. [23] proposed a multivariate ensemble classification (MEC) technique for intrusion detection to enhance the cybersecurity of cyber-physical energy systems (CPES). Their method focused on parameter detection accuracy, computational efficiency, and stability. Ensemble learning has been developed to aggregate the results. Their results show that the MEC method demonstrated promising potential in energy applications. On the other hand, the efficiency of

programmable IDS to protect distributed energy resources (DER), such as Solar and Wind renewable systems, against malicious intrusions was studied [24]. The DER network is connected to an electrical microgrid which makes the whole system susceptible to attacks. The authors claim that through their method, attacks can be detected and mitigated which expands the security of the microgrid-DER system.

Another study proposed an application that employs the SVM method and AN method to detect intrusion rates [25]. Each algorithm is used to determine whether the requested data is legal or contains any errors. They conclude that the ANN algorithm outperforms the SVM approach in terms of intrusion detection. A novel IDS approach based on hybrid sampling and the deep hierarchical network has been suggested and described [26]. They begin by combining one-sided selection (OSS) and Synthetic Minority Oversampling Technique (SMOTE) to create a balanced dataset for model training. It can cut the model's training time to some extent and solve the problem of insufficient training from unbalanced data. Chuang and Li [27] introduced a new hybrid ML technique combining Naive Bayes and C4.5 to improve the developed classification model performance as well as shorten training time in network intrusion detection. A potential intelligent IDS appears by Iwendi et al. [28]. On the NSLKDD and KDD99 datasets, various ensemble machine-learning algorithms were put into practice for testing. However, because the data balancing mechanism was not used, certain attacks have 0 categorization accuracy. This led us to the idea of combining random forest with ensemble learning. This combination will attempt to generate the best decision tree from the random forest. Then, bagging assembling will improve stability and accuracy.

3. METHODOLOGY

In the past 15 years, numerous approaches to building and integrating multiple classifiers have been widely used [29]. This research proposes using ensemble ML with random forest as an approach to build the optimal classifier of possible attacks. Ensemble learning includes bagging and stacking. We applied and tested the proposed method on three datasets to ensure its reliability and transparency. To prepare the reader for a clear understanding of how our proposed method works, the following subsections will begin by describing the dataset used in the experiment. We will then explain the proposed BE-RF methodology in detail.

3.1 Datasets

3.1.1 UNSW-NB15 dataset

UNSW-NB15 dataset [30] was used to evaluate the effectiveness of the proposed methodology. The dataset includes nine different current cyberattack kinds: Analysis,

Backdoors, Denial of Service (DoS), Exploits, Fuzzers, Generic, Reconnaissance, Shellcode, and Worms. This dataset contains 175,341 training records and 82,332 records for testing. Preprocessing includes replacing missing values using K-NN and setting roles to the label, prediction, and regular attributes applied to make the dataset ready for the proposed methodology. The attack category was set to be the label attribute, which is necessary for the classification process.

3.1.2 NSL-KDD dataset

NSL-KDD Datasets feature logs of online behavior seen by a basic IDS and are the phantoms of the activity experienced by genuine IDS and fair the follows of its presence leftovers. The dataset covers 43 highlights per record, with 41 of the highlights alluding to the activity input itself, and the final two are names (either an ordinary or attack) and Score (the seriousness of the activity entered). There are four distinct attack categories: DoS, Probe, User to Root (U2R), and Remote to Local (R2L). An evaluation by Revathi illustrates that these datasets are very proper for matching against other IDS models [31]. These 4 classes are divided into subclasses as Figure 2 illustrates the description of the NSL-KDD dataset [32].

Classes:	DoS	Probe	U2R	R2L
Sub-Classes:	• apache2 • back • land • neptune • mailbomb • pod • processtable • smurf • teardrop • udpstorm • worm	• ipsweep • mscan • nmap • portsweep • saint • satan	• buffer_overflow • loadmodule • perl • ps • rootkit • sqlattack • xterm	• ftp_write • guess_passwd • httptunnel • imap • multihop • named • phf • sendmail • Snmpgetattack • spy • snmpguess • warezclient • warezmaster • xlock • xsnoop
Total:	11	6	7	15

Figure 2. NSL-KDD attack categories

3.1.3 CICIDS2017 dataset

The CICIDS2017 dataset [33] comprises the most current and common benign attacks, closely reflecting genuine real-world data (PCAPs). It also includes the findings of a CICFlowMeter network traffic analysis, which classifies flows depending on the time stamp, source and destination IP addresses, source and destination ports, protocols, and attack. Table 1 explains the names of files, days of activity, the number of samples in each file, and finally, the attacks found. A detailed analysis of this dataset was done [34].

3.2 Random Forest algorithm

A random forest is a collection of a specific statistic of arbitrary trees specified by the number of trees. A decision tree could be a hierarchical collection of nodes aiming to form an option on values aligned to a lesson or an evaluation of arithmetic point value estimation. Each core addresses a specific quality sub-rule. As a classification, this rule leaves values belonging to a particular class. As a fallback, separate them to reduce error in the ideal way for the chosen parametric model. Construction of the modern core is repeated until employment standards are met. A forecast for the course named Trait is decided depending on the majority of cases that come to end point amid era, whereas an evaluation for the arithmetic mean value is obtained by averaging the values at that endpoint. These trees are built/trained on a bootstrap section of the training set given at the training phase. Each core of a tree symbolizes a rule. The resulting rule isolates values in an ideal method for the selected parameter measure. For classification, the rule isolates values that uniquely represent distinctive classes, whereas for regression, it isolates them to decrease the mistake done by the evaluation. Resulting endpoints are constantly formed and rehashed until the halting criteria are encountered.

Table 1. CICIDS2017 general description

Name of Files	Day Activity	Attacks Found
Monday-WorkingHours.pcap_ISCX.csv	Monday	Benign (Normal human activities)
Tuesday-WorkingHours.pcap_ISCX.csv	Tuesday	Benign, FTP-Patator, SSH-Patator
Wednesday-WorkingHours.pcap_ISCX.csv	Wednesday	Benign, DoS GoldenEye, DoS Hulk, DoS Slowhttptest, DoS slowloris, Heartbleed
Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX.csv	Thursday	Benign, Web Attack - Brute Force, Web Attack – Sql Injection, Web Attack - XSS
Thursday-WorkingHours-Afternoon-Infiltration.pcap_ISCX.csv	Thursday	Benign, Infiltration
Friday-WorkingHours-Morning.pcap_ISCX.csv	Friday	Benign, Bot
Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX.csv	Friday	Benign, PortScan
Friday-WorkingHours-Afternoon-DDoS.pcap_ISCX.csv	Friday	Benign, DDoS

Note: DoS: Denial of Service; DDoS: Distributed Denial of Service

Random Forest RF algorithm has been used widely in different scientific fields [35] for a network IDS to classify network risks. The RF system, which aggregates DTs to boost the performance of an individual DT, was developed in 1995 [36]. RFs are made up of DTs on training datasets chosen at random; in this method, forecasts are generated from every tree, and the best answer is chosen using ensemble methods. To lessen the variation of prediction accuracy, the RF algorithm builds *n* DTs on a set of randomly chosen data points and then harmonizes the classification result from every DT (basic classifier) by group voting. Simply put, an RF is a technique for ensemble learning that performs better than a single DT since it averages the DT findings to lessen the effects of overfitting, as illustrated by Figure 3. The importance of a feature is assessed by dividing the weighted difference in node impurity by the probability of accessing that node [37]. By dividing the total of examples by the number of samples that hit the node, the node likelihood can be calculated. With increasing value, the feature becomes more important.

Gini Importance was used to determine a node's significance, with a binary tree considering only two child endpoints:

$$ni_j = w_j C_j - W_{left(j)} C_{left(j)} - W_{right(j)} C_{right(j)} \quad (1)$$

sig(*j*) = the significance of node *j*.
w(*j*) = Sample count nearing node *j* in terms of values.
C(*j*) = node *j*'s imperfection rate.
left(*j*) = divided on node *j* by a left sub-node.
right(*j*) = divided on node *j* by a right sub-node.

The following formula is used to determine each feature's relevance in a decision tree:

$$fi_i = \frac{\sum_{j:\ \text{node } j \text{splits of feature } i} n\, i_j}{\sum_{k \in \text{ all nodes}} n\, i_k} \quad (2)$$

fi_i = the significance of attribute *i*.
ni_j = the significance of node *j*.

A rate between 0 and 1 is applied for normalization purposes by dividing by the sum of all component importance values.

$$\text{norm } fi_i = \frac{fi_i}{\sum_{j \in \text{ all features}} fi_i} \quad (3)$$

The final random forest level feature importance is the average of all trees. The sum of feature importance values for each tree is computed and divided by the entire count of trees.

$$RFfi_i = \frac{\sum_{j \in \text{ all trees}} \text{norm } fi_{ij}}{T} \quad (4)$$

- *RFfi*(*i*) = Compute the significance of element i from all trees in a random forest model.
- norm *fi*(*ij*) = the element significant normalization for *i* in tree *j*.
- *T* = total number of trees.

3.3 BE-RF based Intrusion Detection System

RF ensures the integration of multiple generated decision trees into a single tree with the highest accuracy. On the other hand, bagging ensemble learning (BE) is used to reduce variance and prevent overfitting. This makes the combination of them ideal for high-variance models such as unpruned decision trees.

As illustrated by Figure 3, we used the BE algorithm that pursues searching for robust prediction performance by producing multiple random forest models on different samples of the same dataset and then generating the average for all these produced models to be considered as the suggested optimal prediction. Bootstrapping bagging ensemble learning algorithm tries to expand the classification accuracy with reference to accuracy and stability. It further helps diminish variance and bypass overfitting.

3.3.1 Data preprocessing

For evaluation purposes, we used three freely available datasets for IDS, which have received considerable attention in previous studies. Any work with datasets requires preliminary preprocessing steps. The first step was to process the unbalanced data to produce balanced data, and we applied two techniques as previously suggested by Jiang et al. [26]. Initially, noise samples in the majority class are eliminated using the OSS technique. Second, we use the SMOTE algorithm to create minority class samples to correct the imbalance in the network traffic data. The dataset is refined in the first phase to only take samples from the minority class into consideration. Following that, a search on the *K* closest neighbors, including all samples, is carried out. The system then chooses a random sample for this sample, along with a random nearest neighbor. Just on the line separating the two samples, a new sample is produced. As a result, imbalanced data can become balanced data.

To produce the random forest tree, several factors must be specified: Gain_ratio: A variation of data pick up that alters the data pick up for each Quality to permit the breadth and

consistency of the Trait values. Maximal profundity: The profundity of a tree shifts based on measure and properties of the set. This criterion is utilized to stop profundity. The maximum depth for this study was set to 5 to ease reaching the fast leaf (decision). Confidence: This parameter indicates the certainty level utilized for the cynical mistake calculation of pruning.

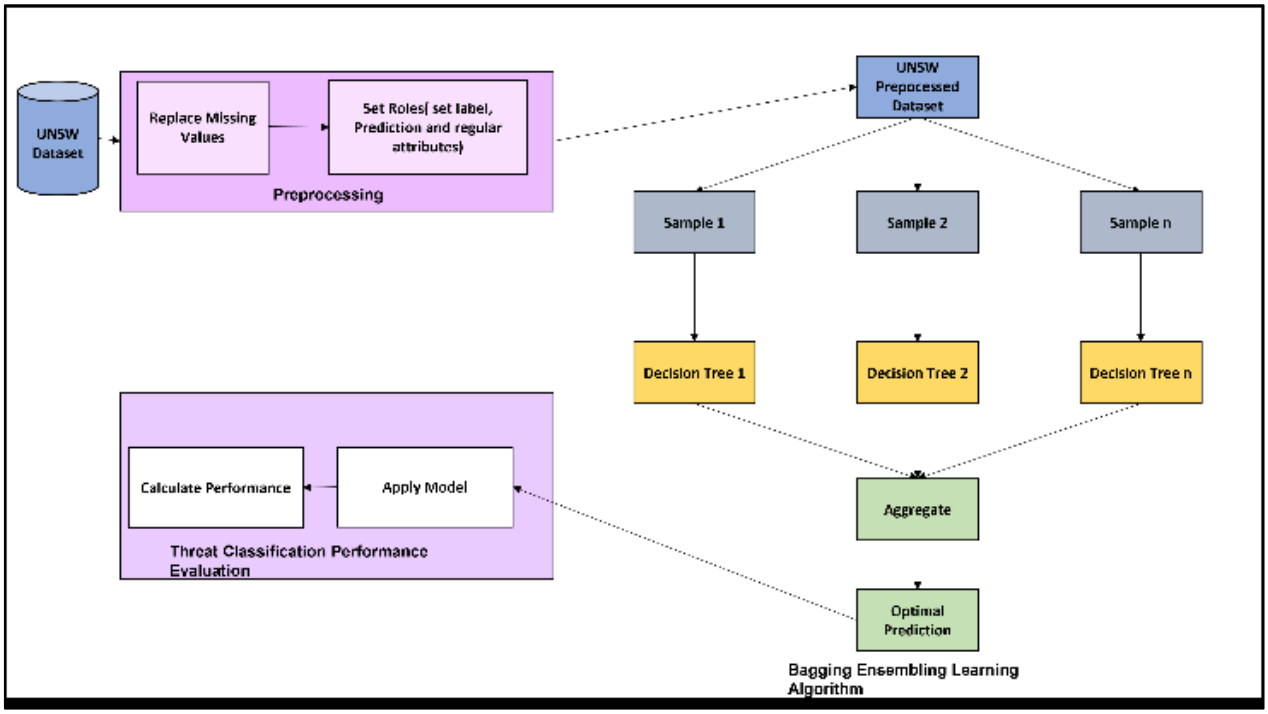

Figure 3. Overall architecture of the proposed Intrusion Detection System (IDS)

Algorithm 1: BE-RF for IDS

1 Input: UNSW-NB15 OR NSL-KDD OR CICIDS2017 Datasets
2 Set I to be the maximum number of iterations of the bagging ensemble learning algorithm
3 Begin loop
4 **For *i*=1 to I do**
5 Generate bagging samples
6 Set the decision tree criteria (gain-ratio, maximum depth, confidence)
7 Train the random forest tree on the bagging sample
8 Calculate the optimal prediction function: $\widehat{PF} = \widehat{pf_1(x)} + \widehat{pf_2(x)} + \cdots + \widehat{pf_n(x)} = \sum_{i=1}^{n} \widehat{pf_i(x)}$ PF= *PF*=ensembled bagged function While *pf(x)* are the individual learners
9 Draw the random forest trees and report the associated rules
10 **End for loop**

3.3.2 Implementation

The preprocessed dataset is then forming the input to the BE-RF algorithm which is a nested setup with sub-processes. The main sub-process is the learner, which requires a training dataset to generate a model. Providing the sub process learners, this algorithm strives to develop an optimal classification model to help the IDS classify the suspected attack behaviors into its attack types. Ensemble strategies utilize different models to get distant better; much better; higher; stronger; and improved; a Better predictive execution than any alternative model. Alternatively, an ensemble may be a procedure for combining numerous powerless learners in an endeavor to create a solid learner. Assessing the forecast of an outfit ordinarily requires more computation than assessing the forecast of a single show, so an ensemble may be thought of as a way to compensate for limited learning calculations by performing a part of additional computation.

An ensemble is a directed learning calculation since it can be prepared and then utilized to form expectations. The trained ensemble, subsequently, speaks to a single speculation. This theory is radically not included in the speculative space of the models in which it is built. This allows the outfits to appear more adaptive. This adaptability could, hypothetically, allow us to customize more preparation information than a single demonstration; still, a few outfit strategies (particularly stowing) tend to decrease issues related to the over-fitting of the preparation information. The concept of ensemble bagging learning is used to aggregate predictions from compound classification models, or from the matching category of models for distinctive learning datasets. It is also used to cope with the naturally ambiguous results that arise when adopting compound models to narrow datasets. Exceptionally distinctive trees will frequently be developed for the diverse tests, outlining the insecurity of models regularly apparent with small information sets. One strategy for determining a single forecast (for unused perceptions) is to utilize all trees found within the diverse tests, in addition to employing straightforward voting: The ultimate prediction is the highest regularly anticipated by the distinctive trees. Finally, the developed model is to be evaluated.

A model is first trained on a training set via a learning algorithm. Afterward, the trained output would be applied to other data, which is called a testing set. Usually, the goal is to get classifications on undetected data or try pre-trained models to transform the data. The performance of the developed classification model is to be discussed in the following section.

4. EXPERIMENTAL RESULTS

We used the RapidMiner platform to build and assess the suggested method's classification model [38]. The split ratio was 70% for training and 30% for testing with a stratified sampling mechanism. Stratified sampling ensures that the population is divided into non-overlapping, homogeneous subgroups and the final sample accurately represent these key groups. Missing values were handled using the Replace-Missing-values operator while setting the attribute-filter-type to all and the default values to be the average. For the benchmarking of BE-RF with previous methods on the three datasets, we used accuracy, precision, recall, and F1 measurement. Precision, recall, accuracy, and F1-measure are the most famous performance metrics for any classification task. These performance metrics are calculated as follows:

$$Precision = TP(TP + FP) \quad (5)$$

$$Recall = TP(TP + FN) \quad (6)$$

$$Accuracy = TP + TN/(TP + TN + FP + FN) \quad (7)$$

$$\text{F1} - \text{Measure} = (2 * Precision * Recall)/(precision + recall) \quad (8)$$

where, TP = True positive means the number of true attack records correctly classified as true attacks, FP = False positive means the number of ordinary records falsely classified as an attack, TN = True negative means the number of ordinary records correctly classified as ordinary records, and FN = False negative means the number of attack records falsely classified as ordinary records.

Numerous ML and deep learning algorithms are used to build classification models for IDS systems. For Instance, Random Forest and classic convolutional neural systems are broadly utilized in IDS discovery. Hence, the classification algorithms commonly utilized in IDS are compared with the proposed methodology presented in this paper. RF, AlexNet, LeNet-5, CNN, BiLSTM, and CNN-BiLSTM systems are utilized in this paper for the comparison of classification tasks and performance.

4.1 Experimental results on UNSW-NB15 dataset

Table 2 illustrates that the proposed methodology of using the BE-RF approach scores better classification accuracy, precision, recall, and F1 compared with other classifiers reported in the study by Jiang et al. [26] when applied to the UNSW-NB15 dataset. For better clarification, Figure 4 proves that the BE-RF approach triggered using random forest provides a better classification accuracy against the previous works produced by different studies. This can be explained by BE's ability to successfully promote the selected decision tree that was chosen among several decision trees generated by the random forest process.

Table 2. BE-RF performance against various classifiers on UNSW-NB15 dataset

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Measure (%)
Bagging Ensemblelearning	86.03	94.85	87.62	90.43
CNN-BiLSTM	77.16	82.63	79.91	81.25
BiLSTM	72.24	79.52	72.43	75.81
CNN	74.61	81.01	75.65	78.24
LeNet-5	71.11	78.87	71.13	74.80
AlexNet	73.89	79.61	72.73	76.01
RF	73.46	80.84	73.64	77.07

Note: CNN: Convolutional Neural Network; BiLSTM: Bidirectional Long Short-Term Memory network; RF: Random Forest

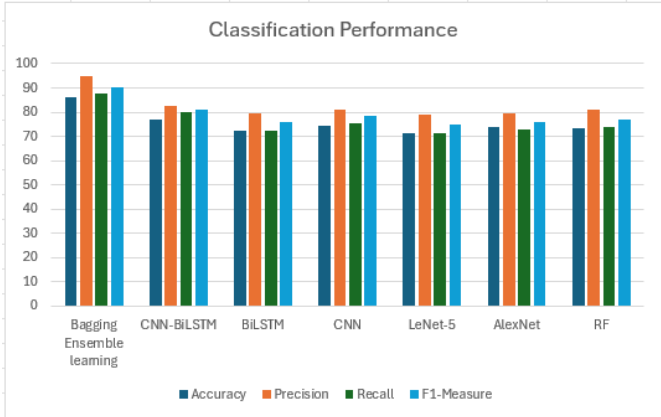

Figure 1. BE-RF classification outperforms other classifiers on UNSW-NB15

4.2 Experimental results on NSL-KDD dataset

Table 3. Classification precision and recall on NSL-KDD different classes

	True Normal	True Dos	True R2L	True Probe	True U2R	Class Precision (%)
Pred. normal	22956	121	320	239	24	97.02
Pred. Dos	50	15891	6	22	3	99.49
Pred. R2L	45	0	828	5	3	93.98
Pred. Probe	65	4	9	3957	0	98.07
Pred. U2R	0	0	1	0	6	85.71
Classre-call	99.31%	99.22%	71.13%	93.70%	16.67%	

Note: DoS: Denial of Service; R2L: Remote to Local; U2R: User to Root

Table 4. BE-RF classification metrics on NSL-KDD dataset

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Measure (%)
Bagging Ensemble Learning	97.9	94.86	76.01	84.3
CNN-BiLSTM	83.58	85.82	84.49	85.14
BiLSTM	79.43	81.14	79.65	80.39
CNN	81.75	82.43	82.71	82.57
LeNet-5	79.91	82.95	80.01	80.45
AlexNet	77.02	78.54	77.24	77.88
RF	74.71	81.33	75.49	78.30

Note: CNN: Convolutional Neural Network; BiLSTM: Bidirectional Long Short-Term Memory network; RF: Random Forest

Each experiment consists of two parts: training and testing. During the training, a classification model of the attack category is formed. In the testing part, the generated model is tested against a new example set to test and validate the accuracy and performance of the classification model. These two steps are essential as they give a comprehensive performance evaluation of the developed classification model to assist in deploying this model to the real environment. The classification performance metrics for each attack category on the NSL-KDD dataset are shown in Table 3. It can be seen that

BE-RF performs well in classifying all the attack categories except for U2R category due to the fact that the example set for this category was so small compared to the other categories. Due to that, the classifier did not get the chance to learn enough information on such attack categories during training. Table 4 and Figure 5 illustrate that BE-RF performs better than the previous classifiers in terms of accuracy and precision.

4.3 Experimental results on CICIDS2017 dataset

This part gives a rigorous assessment of the suggested system's performance on CICIDS2017 dataset. It is worth mentioning that CICIDS2017 datasets are dispersed among eight files. Each file is labeled with the day and period when the attacks were detected. Therefore, we evaluated BE-RF initially for each file as illustrated by Table 5, then combined all the files and recalculated the classification accuracy.

We found that processing each file was an extremely tedious task. Thus, we merged those files to create one large file. Furthermore, merging them into a single file is crucial, as we observed that each file contained specific types of attacks that might not be present in the others. Merging the files into one allows us to address all types of potential threats and breaches within a single file. After merging the files, we discovered that the combined dataset provides information on every possible modern attack classification in one place. However, the size of the combined dataset grows exponentially at the same time. This huge amount of data becomes a shortage in itself. The disadvantage is that more overheads are used for fetching and computing. Therefore, we believe that using BE will help with data sampling, reducing the time and computational effort required. The merged CICIDS2017 file has 288602 cases with missing class labels and 203 instances with missing metadata.

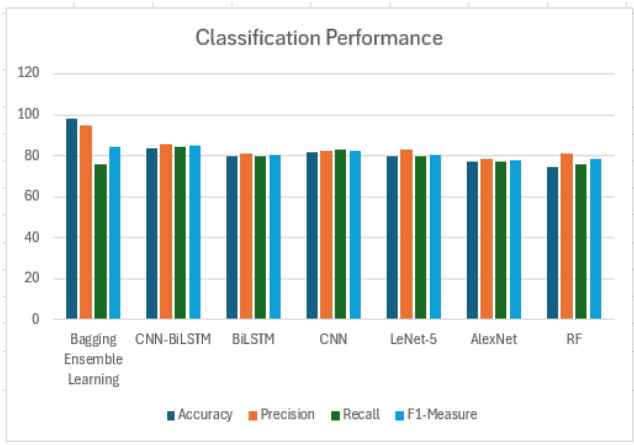

Figure 5. Visualization of BE-RF classification benchmarking on NSL- KDD

Table 5. BE-RF performance evaluation on the CICIDS2017 files

File Name	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
Tuesday-WorkingHours.pcap_ISCX.csv	99.32	99.77	81.27	89.57
Wednesday-workingHours.pcap_ISCX.csv	99.37	65.30	85.6	80
Thursday-WorkingHours-Morning-WebAttacks.pcap_ ISCX.csv	99.48	66.97	84.54	75
Thursday-WorkingHours-Afternoon-Infilteration.pcap_ ISCX.csv	99.98	99.89	67.00	80
Friday-WorkingHours-Morning.pcap_ISCX.csv	99.22	94.62	79.56	86
Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX.csv	99.91	99.91	99.91	99.91
Friday-WorkingHours-AfternoonDDos.pcap_ISCX.csv	99.85	99.85	99.85	99.85

Table 5. BE-RF performance comparison on CICIDS2017 dataset

Classifier	Accuracy (%)	F-Measure (%)
BE-RF	99.99	99
KODE	99.99	99
EM	96.34	96.3
DBSCAN	98.76	98.7
One-Class SVM	97.92	97.9
K-mean	99.92	99.2

Note: SVM: Support Vector Machines

The performance of the proposed BE-RF methodology was compared with a range of recent methodologies, which also demonstrated good performance, such as K-means, One-Class SVM KODE, DBSCAN, EM, and the merge of the four previous algorithms in one ensemble learning called KODE [12]. Table 6 shows the performance comparison of BE-RF against those classifiers. The KODE method showed high performance when it combined the performance of four algorithms. However, our proposed methodology BE-RF showed a higher ability to classify the type of attacks and thus early protection in terms of accuracy, precision, recall and F1 measure. This is due to the steps taken, starting with the handling of unbalanced and incomplete data, and then to the effort made by the random forest algorithm in building confidence, which describes the level of confidence used in calculating the wrong pruning that it chooses as a path to reach the goal.

A sample decision tree created by the BE-RF application on CICIDS2017 can be shown in Figure 6. BE helps RF by training multiple models on different subsets of data (bootstrap samples) and averaging their results, often reducing variance. Such tree representation is important to draw a road map to understand the pattern of possible attacks. Various rules can be generated from those trees to be followed intelligently by the IDS to detect any suspicious connection.

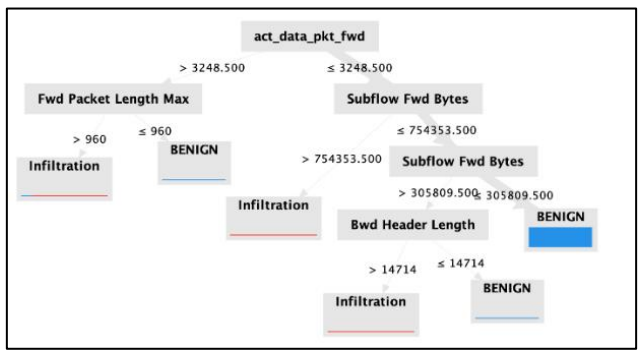

Figure 6. BE-RF generated trees for CICIDS2017 Thursday attacks file with root=act_data_pkt_fwd

Although BE-RF outperformed various classifiers on those datasets, we found that the method requires high consumption

of computing resources and memory. This is because Bagging requires training and storing multiple independent base models (often 100+ decision trees). This limitation can be an issue for future work.

5. CONCLUSIONS

Cyber-attacks pose a major challenge to all sectors of society, especially with the large and accelerating transformation of the use of technology to facilitate people's lives, especially after the Covid-19 pandemic. Even the advanced critical infrastructure, such as smart grids and power plants are not immune to these cyber threats. Researchers seek to find appropriate solutions to detect cyber-attacks in order to thwart them. One of the most important technologies used recently is the inclusion of AI, especially ML, to understand the patterns of cyber-attacks and detect them quickly. In this paper, we propose using the Bagging Ensemble learning method to develop an accurate classification model that is able to classify the network traffic into its possible attack patterns. We employed UNSW-NB15, NSL-KDD, and CICIDS2017 datasets to evaluate the performance of the proposed methodology (BE-RF) against several previously used classifiers, namely, RF, AlexNet, CNN, and BiLSTM. The ensemble nature of Random Forest helps to improve overall detection accuracy for both binary (normal/attack) and multi-class classification. Results show that the proposed Bagging Ensemble Learning Fostered by a random forests classifier, performs better than all other classifiers. One main limitation we found is that some datasets are static, which often lack the characteristics of evolving modern intrusion types. This opens the door for future research to explore the use of hybrid ensemble approaches as an adaptive classifier that can work on new traffic types.

ACKNOWLEDGMENT

The authors are thankful to the Deanship of Graduate Studies and Scientific Research at University of Bisha for supporting this work through the Fast-Track Research Support Program.

REFERENCES

[1] Muhammad, A.R., Sukarno, P., Wardana, A.A. (2023). Integrated security information and event management (SIEM) with intrusion detection system (IDS) for live analysis based on machine learning. Procedia Computer Science, 217: 1406-1415. https://doi.org/10.1016/j.procs.2022.12.339

[2] Satilmiş, H., Akleylek, S., Tok, Z.Y. (2024). A systematic literature review on host-based intrusion detection systems. IEEE Access, 12: 27237-27266. https://doi.org/10.1109/ACCESS.2024.3367004

[3] Kizza, J.M. (2024). System intrusion detection and prevention. In Guide to Computer Network Security (pp. 295-323). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-031-47549-8_13

[4] Toldinas, J., Venčkauskas, A., Damaševičius, R., Grigaliūnas, Š., Morkevičius, N., Baranauskas, E. (2021). A novel approach for network intrusion detection using multistage deep learning image recognition. Electronics, 10(15): 1854. https://doi.org/10.3390/electronics10151854

[5] Wang, A., Wang, W., Zhou, H., Zhang, J. (2021). Network intrusion detection algorithm combined with group convolution network and snapshot ensemble. Symmetry, 13(10): 1814. https://doi.org/10.3390/sym13101814

[6] Imran, Jamil, F., Kim, D. (2021). An ensemble of prediction and learning mechanism for improving accuracy of anomaly detection in network intrusion environments. Sustainability, 13(18): 10057. https://doi.org/10.3390/su131810057

[7] Jaw, E., Wang, X. (2021). Feature selection and ensemble-based intrusion detection system: An efficient and comprehensive approach. Symmetry, 13(10): 1764. https://doi.org/10.3390/sym13101764

[8] Jamil, N., Qassim, Q.S., Bohani, F.A., Mansor, M., Ramachandaramurthy, V.K. (2021). Cybersecurity of microgrid: state-of-the-art review and possible directions of future research. Applied Sciences, 11(21): 9812. https://doi.org/10.3390/app11219812

[9] An, L., Yang, G.H. (2022). Enhancement of opacity for distributed state estimation in cyber–physical systems. Automatica, 136: 110087. https://doi.org/10.1016/j.automatica.2021.110087

[10] An, L., Yang, G.H. (2019). Distributed secure state estimation for cyber–physical systems under sensor attacks. Automatica, 107: 526-538. https://doi.org/https://doi.org/10.1016/j.automatica.2019.06.019

[11] Pasqualetti, F., Dörfler, F., Bullo, F. (2013). Attack detection and identification in cyber-physical systems. IEEE Transactions on Automatic Control, 58(11): 2715-2729. https://doi.org/10.1109/TAC.2013.2266831

[12] Nassrullah, K.S., Stepanyan, I.V., Nasrallah, H.S., Florez, N.J.M., Zidoun, A.M., Mohammed, S.R. (2024). Unsupervised machine learning control techniques for solving the general synthesis of control system problem. International Journal of Intelligent Engineering & Systems, 17(3): 401-416. https://doi.org/10.22266/ijies2024.0630.32

[13] Krause, T., Ernst, R., Klaer, B., Hacker, I., Henze, M. (2021). Cybersecurity in power grids: Challenges and opportunities. Sensors, 21(18): 6225. https://doi.org/10.3390/s21186225

[14] Sridharan, S., Patil, S., Shobha, T., Pai, P. (2025). Hybrid machine learning–based intrusion detection for zero-day attack prevention in digital education networks. International Journal of Safety & Security Engineering, 15(8): 1703-1713. https://doi.org/10.18280/ijsse.150815

[15] Dou, J., Yunus, A.P., Bui, D.T., Merghadi, A., Sahana, M., Zhu, Z., Pham, B.T. (2020). Improved landslide assessment using support vector machine with bagging, boosting, and stacking ensemble machine learning framework in a mountainous watershed, Japan. Landslides, 17(3): 641-658. https://doi.org/10.1007/s10346-019-01286-5

[16] Srivastava, A., Agarwal, A., Kaur, G. (2019). Novel machine learning technique for intrusion detection in recent network-based attacks. In 2019 4th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India, pp. 524-528. https://doi.org/10.1109/ISCON47742.2019.9036172

[17] Singhal, A., Maan, A., Chaudhary, D., Vishwakarma, D. (2021). A hybrid machine learning and data mining based approach to network intrusion detection. In 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), Coimbatore, India, pp. 312-318. https://doi.org/10.1109/ICAIS50930.2021.9395918

[18] Chen, L., Kuang, X., Xu, A., Suo, S., Yang, Y. (2020). A novel network intrusion detection system based on CNN. In 2020 Eighth International Conference on Advanced Cloud and Big Data (CBD), Taiyuan, China, pp. 243-247. https://doi.org/10.1109/CBD51900.2020.00051

[19] Yang, Y., Zheng, K., Wu, C., Niu, X., Yang, Y. (2019). Building an effective intrusion detection system using the modified density peak clustering algorithm and deep belief networks. Applied Sciences, 9(2): 238. https://doi.org/10.3390/app9020238

[20] Moustafa, N., Slay, J. (2015). UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, ACT, Australia, pp. 1-6. https://doi.org/10.1109/MilCIS.2015.7348942

[21] Alalade, E.D. (2020). Intrusion detection system in smart home network using artificial immune system and extreme learning machine hybrid approach. In 2020 IEEE 6th World Forum on Internet of Things (WF-IoT), New Orleans, LA, USA, pp. 1-2. https://doi.org/10.1109/WF-IoT48130.2020.9221151

[22] Egger, M., Eibl, G., Engel, D. (2020). Comparison of approaches for intrusion detection in substations using the IEC 60870-5-104 protocol. Energy Informatics, 3(Suppl 1): 15. https://doi.org/10.1186/s42162-020-00118-4

[23] Li, Y., Xue, W., Wu, T., Wang, H., Zhou, B., Aziz, S., He, Y. (2021). Intrusion detection of cyber physical energy system based on multivariate ensemble classification. Energy, 218: 119505. https://doi.org/10.1016/j.energy.2020.119505

[24] Ma, S., Li, Y., Du, L., Wu, J., Zhou, Y., Zhang, Y., Xu, T. (2022). Programmable intrusion detection for distributed energy resources in cyber–physical networked microgrids. Applied Energy, 306: 118056. https://doi.org/10.1016/j.apenergy.2021.118056

[25] Yedukondalu, G., Bindu, G.H., Pavan, J., Venkatesh, G., SaiTeja, A. (2021). Intrusion detection system framework using machine learning. In 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, pp. 1224-1230. https://doi.org/10.1109/ICIRCA51532.2021.9544717

[26] Jiang, K., Wang, W., Wang, A., Wu, H. (2020). Network intrusion detection combined hybrid sampling with deep hierarchical network. IEEE Access, 8: 32464-32476. https://doi.org/10.1109/ACCESS.2020.2973730

[27] Chuang, P.J., Li, S.H. (2019). Network intrusion detection using hybrid machine learning. In 2019 International Conference on Fuzzy Theory and Its Applications (iFUZZY), New Taipei, Taiwan, pp. 1-5. https://doi.org/10.1109/iFUZZY46984.2019.9066223

[28] Iwendi, C., Khan, S., Anajemba, J.H., Mittal, M., Alenezi, M., Alazab, M. (2020). The use of ensemble models for multiple class and binary class classification for improving intrusion detection systems. Sensors, 20(9): 2559. https://doi.org/10.3390/s20092559

[29] Kittler, J., Roli, F. (2009). Multiple classifier systems. In 8th International Workshop, MCS 2009, Reykjavik, Iceland. https://doi.org/10.1007/978-3-642-02326-2

[30] Moustafa, N. (2017). Designing an online and reliable statistical anomaly detection framework for dealing with large high-speed network traffic (Doctoral dissertation, UNSW Sydney), Australia.

[31] Revathi, S., Malathi, A. (2013). A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection. International Journal of Engineering Research & Technology (IJERT), 2(12): 1848-1853.

[32] Publish AI, ML & data-science insights to a global community of data professionals. Towards Data Science. https://towardsdatascience.com/a-deeper-dive-into-the-nsl-kdd-data-set-15c753364657.

[33] Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018), 108-116. https://doi.org/10.5220/0006639801080116

[34] Panigrahi, R., Borah, S. (2018). A detailed analysis of CICIDS2017 dataset for designing intrusion detection systems. International Journal of Engineering & Technology, 7(3.24): 479-482.

[35] Lin, H.C., Wang, P., Chao, K.M., Lin, W.H., Yang, Z.Y. (2021). Ensemble learning for threat classification in network intrusion detection on a security monitoring system for renewable energy. Applied Sciences, 11(23): 11283. https://doi.org/10.3390/app112311283

[36] Ho, T.K. (1995). Random decision forests. In Proceedings of 3rd International Conference on Document Analysis and Recognition, Montreal, QC, Canada, pp. 278-282. https://doi.org/10.1109/ICDAR.1995.598994

[37] Ronaghan, S. (2018). The mathematics of decision trees, random forest and feature importance in scikit-learn and spark. Towards Data Science, 11. https://medium.com/data-science/the-mathematics-of-decision-trees-random-forest-and-feature-importance-in-scikit-learn-and-spark-f2861df67e3.

[38] Hofmann, M., Klinkenberg, R. (2016). RapidMiner: Data Mining Use Cases and Business Analytics Applications. CRC Press.