IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 August 2022, accepted 22 August 2022, date of publication 26 August 2022, date of current version 2 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3201869

RESEARCH ARTICLE

MLTs-ADCNs: Machine Learning Techniques for Anomaly Detection in Communication Networks

HAIDER W. OLEIWI[1], DOAA N. MHAWI[2], AND HAMED AL-RAWESHIDY[1], (Senior Member, IEEE)
[1]Department of Electronic and Electrical Engineering, Brunel University London, London UB8 3PH, U.K.
[2]Department of Computer Systems Techniques, Middle Technical University, Baghdad 10010, Iraq

Corresponding author: Hamed Al-Raweshidy (hamed.al-raweshidy@brunel.ac.uk)

ABSTRACT From a security perspective, the research of the jeopardized 6G wireless communications and its expected ultra-densified ubiquitous wireless networks urge the development of a robust intrusion detection system (IDS) with powerful capabilities which could not be sufficiently provided by the existing conventional systems. IDSs are still insufficient against continuous renewable unknown attacks on the wireless communication networks, especially with the new highly vulnerable networks, leading to low accuracy and detection rate with high (false-negative and false-positive) rates. To this end, this paper proposed a novel anomaly detection in communication networks by using an ensemble learning (EL) algorithm-based anomaly detection in communication networks (ADCNs). EL-ADCNs consists of four main stages; the first stage is the preprocessing steps. The feature selection method is the second stage. It adopts the proposed hybrid method using correlation with the random forest algorithm of ensemble learning (CFS–RF). It reduces dimensionality and retrieves the best subset feature of all the three datasets (NSL_KDD, UNSW_NB2015, and CIC_IDS2017) separately. The third stage is using hybrid EL algorithms to detect intrusions. It involves modifying two classifiers (i.e., random forest (RF), and support vector machine (SVM)) to apply them as adaboosting and bagging EL Algorithms; using the voting average technique as an aggregation process. The final stage is testing the proposal using binary and multi-class classification forms. The experimental results of applying 30, 35, and 40 features of the proposed system to the three datasets achieved the best results of a 99.6% accuracy with a 0.004 false-alarm rate for NSL_KDD, a 99.1% accuracy with a 0.008 false-alarm rate for UNSW_NB2015, and a 99.4% accuracy with a 0.0012 false-alarm rate for CIC_IDS2017.

INDEX TERMS Adaboosting algorithm, Bagging algorithm, correlation feature selection, ensemble method, intrusion detection systems.

I. INTRODUCTION

The new era of wireless communications, changeable mobile network infrastructure, the proliferation of connected Internet of Everything (IoE) devices/applications, and the variety of expected services raise critical security concerns and present complications with high risks of data security at the networks' core and edge. Due to the high vulnerability of communication networks to various renewable attacks, academic and industrial research must prioritize deploying intelligent security systems to satisfy the emerging requirements and technologies of the next generations of communication systems (6G and beyond). Thereby, it is critical to develop a robust intrusion detection system (IDS) to eliminate those risks sufficiently other than the existing insufficient security systems that cannot adapt to the updatable attacks [1].

The networks' edges connect several types of billions of served nodes that provide various services, e.g., communicating, computing, processing, or sensing for multiple applications via a base station (BS) using terahertz (THz) radio frequency signals [1], [2].

The detection of zero-day attacks is a complex task. Daily, a huge number of suspicious activities are being detected. Whereas the consequences of those complex intrusions are

The associate editor coordinating the review of this manuscript and approving it for publication was Joey Tianyi Zhou.

becoming influential hazards that introduce additional difficulties to the existing IDSs [3], [4], [5], [6], [7], [8], [9], [10].

When IDSs detect unexpected activities or recognized dangers, they issue alerts. An intrusion is any harmful activity that disrupts the information system [11]. IDSs observe computer systems for any odd activity that a traditional packet filter could miss. They scan network packets for signs of potentially harmful behavior, cyber resiliency in defiance of disrupting activities, and illegal access to the system. Signature Intrusion Detection Systems (SIDS) and Anomaly Intrusion Detection Systems (AIDS) are the two methods used by IDSs to identify intrusions [12], [13], [14]. AIDS has flaws and high false-alarm rates [15], [16], [17]. To address these flaws, a novel IDS model that incorporates SIDS and AIDS was provided to improve accuracy and reduce FAR. SIDS could detect common incursions, whereas AIDS could detect new ones [9].

Intrusion detection (ID) is a data analysis in which data mining (DM) techniques used to discover, extract, and distinguish the normal or intrusive patterns automatically. There are four types of tasks typically used in DM: classification, clustering, regression, and association rule learning are all techniques used to learn rules [18], [19]. The feature selection (FS) approach is an essential IDS process to specify the influential features and cancel the worthless features for less performance devolution. [20], [21], [22]. Correlation FS (CFS) uses a correlation-based heuristic evaluation function to rank features. It contrasts the attribute vectors' subsets linked to class-label and not to each other. According to the CFS algorithm, the irrelevant features with minimal link to the class must be omitted. Excess features should be investigated as they are frequently linked to one or more of the other features [23].

The weak learners are models used as a development part of the complex models, merging several weak learners by ensemble learning (EL) methods. For the majority of the time, those essential models are not efficient when they work individually due to the bias (e.g., degree of freedom insufficiencies) or the variation to be dependable (e.g., high degree of freedom). Ensemble approaches aim to decrease weak learners bias/variance, combining a large number of them into a strong learner (i.e., an ensemble algorithm) that performs better [24].

The technological world is moving towards IoE and sophisticated networking based on devices with lightweight algorithms. Despite the continued efforts of researchers, intrusion detection systems still lack the required optimization of detection rate (DR), false alarm rate (FAR), FNR, FPR, or time complexity (execution time) due to the high dimensions of the standard dataset and many Zero-day attacks. Moreover, time complexity has not been considered an influential factor despite its direct impact on resources. This paper provides a proposed method for dimensionality reduction with an FS for extracting the optimal subset of the original features. Then, passing these subsets to the proposed hybrid EL increases the stability and accuracy of the IDS with minimizing the required computation and consequent time.

The proposal trains the FS method and hybrid EL algorithms to attain accurate and efficient IDs. The major contributions of this paper are:

- In the context of FS, we propose a novel method based on CFS combined with forest panelized attributes (CFS-RF) used to assess the correlation of the selected features. It is very beneficial to enhance the efficiency of the training and testing phases.
- We improve the performance of the binary class and the multi-class forms applied to the three unbalanced datasets. The proposal introduces hybrid ensemble algorithms by modifying two various classifiers to work as adaboosting, then combining decisions from multiple ensemble classifiers [random forest (RF) and support vector machine (SVM)] into one decision using the voting average technique (bagging method).

The rest of this paper is structured as follows: Section II presents several related works. The proposed system, methodology, and different machine learning (ML) algorithms are defined in detail in section III. Section IV describes the implementation of the used datasets with the proposed system, while section V discusses the experimental results. Finally, the conclusion and future work are summarized in section VI.

II. RELATED WORK

Dwar Koba, Gaik-wad, and Ravindra Thool proposed "DAREnsemble: Decision tree and rule learner-based ensemble for network intrusion detection system." A new architecture of DAR ensemble was proposed for IDSs that consist of unstable base classifiers using NSL KDD dataset. The experimental results showed 80%, 81%, and 15.1% for accuracy, DR, and FAR, respectively [25].

Hamed, et. al, proposed "Two-tier network anomaly detection model: A machine learning approach" using two-class ML-based classification models, KNN certainty factor voting classifiers where dimensionality reduction was done using linear discriminant analysis. Two generated training datasets used to train the model with SMOTE method for evaluating the selected similarity to deal with the network imbalance of anomaly datasets. The experimental evaluation using NSL-KDD showed an accuracy of 83.24%, FAR of 4.83%, TPR of 82%, and FPR of 5.43 when 16 features were chosen [26].

Kanakarajan and Muniasamy K. proposed "Improving the accuracy of intrusion detection using gar-forest with feature selection:" Those researchers have applied greedy randomized adaptive search procedures with annealed randomness-Forest (GAR-Forest) with FS processes, e.g., information gain, symmetrical uncertainty, feature-subset based on correlation, and NSL-KDD datasets. The results showed an accuracy of 85.0559% with 32-features for binary class and information gain achieved an accuracy of a 78.9035% with 10-features for multi-class [27].

Mittal, et. al, suggested: "Machine learning techniques for energy efficiency and anomaly detection in hybrid wireless sensor networks" using NSL KDD to detect the attack on the wireless sensor network. The experimental results showed that accuracy was 95%, whereas precision, recall, and F1-Score were 94.00%, 98.00%, and 96.00%, respectively [28].

Jaw and Wang proposed "FS and EL-IDS: An Efficient and Comprehensive technique:" A genetic algorithm-based FS methodology and EL algorithm-based logistic regression for NIDS. The results showed 98.99%, 98.73%, and 97.997% of accuracy with 98.75%, 96.64%, and 98.93% of detection rates for CIC_IDS2017, NSL_KDD, and UNSW_NB2015, respectively, using 11, 8, and 13 features [29].

N. Gupta, et. al, suggested the CSE-IDS using cost-sensitive deep learning (DL) with ensemble algorithms to treat an imbalanced class of IDSs. It consists of 3 phases; phase 1 uses a deep neural network (DNN) to divide and disseminate normal or suspicious network attacks. In phase 2, eXtreme Gradient Boosting is used to classify main attacks. However, for phase 3, RF is adopted for minor attacks' classification. The researchers adopted NSL_KDD, CIDDS-001, and CIC_IDS2017 datasets for system performance evaluation, while the accuracies were 99%, 96%, and 92% for NSL, CIDDS-001, and CIC_IDS2017, respectively, whereas the complexity time measurement has taken several hours [30].

In [31], Mighan and Kahani have adopted a stacked auto encoding network to extract features. Afterward, they proposed random forest, SVM, and another classifying method.

Souza *et al.* [32] have presented a DNN-KNN hybrid binary classifying méthodologies. There were a number of hybrid ML and DL algorithms.

Doaa, Ammar, and Soukeana in [33] have adopted feature selection (i.e., correlation feature selection-forest attribute) and ensemble learning techniques. The experimental result of this work used only the CIC_IDS2017 dataset. Furthermore, the testing accuracy reached 87% using 30 feature-selected.

In [34], Doaa and Soukeana have proposed correlation feature selection methods to select the best feature by applying only two datasets (i.e., NSL-KDD and UNSW_BN2015). Moreover, they have chosen only 30 features for those datasets.

Several researchers have studied distributed ML algorithms [35], [36], and they treat high dimensional data in a considerably short time and sufficiently. They have shown the benefits of using them to deal with massive data for preprocessing stage of IDS. Whereas, in the multiple target anomaly classifying step, DL algorithms could reach hidden features to detect unknown attacks while.

To the best of the author's knowledge, the presented system achieves the best results and the highest performance compared to previous systems. It outperforms the state-of-the-art performance using multiple datasets and significantly achieves the best detection, false-alarm, and false-negative rates, in addition to the lowest complexity time.

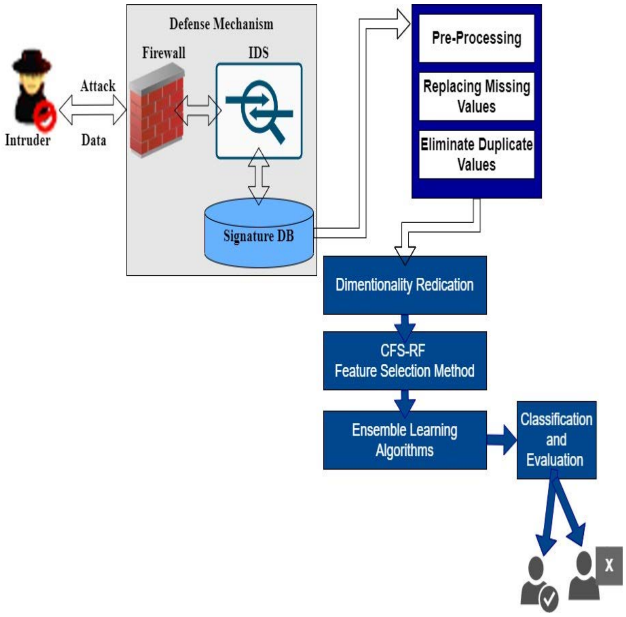

Defense Mechanism
Firewall
IDS
Attack
Intruder
Data
Signature DB
Pre-Processing
Replacing Missing Values
Eliminate Duplicate Values
Dimentionality Redication
CFS-RF Feature Selection Method
Ensemble Learning Algorithms
Classification and Evaluation
X

FIGURE 1. The general structure of anomaly detection in communication networks.

III. METHODOLOGY

A. OVERVIEW

IDS can track malicious activity over the entire network. It was introduced into a wireless communication network to verify any unusual activity during control transmission and data transmission. The intruder tries to attack the network to block transmission or steal precious information from the networks. The intruder embeds bugs into the networks by breaking the network security and unbalancing the activities in the sensor network. In order to overcome this problem, a robustly secured framework is required to save the system from malicious attacks. Figure 1 shows the proposed framework's general structure.

The proposed framework consists of various steps to detect anomalies. First, the defense mechanism consists of an IDS system with databases that position behind the firewall (i.e., data collected from the network, which undergoes preprocessing). After preprocessing, it needs to detect the missing values in the system and then replace the null values with other values. By default, average values are considered, then, duplicate values are removed from the dataset. The encoded data goes through a dimensional reduction process to help with data handling. Thus, feature optimization is done to fetch the optimal features from the data, which assists anomaly detection. Further, the cleaned data is passed to the next stage to select only the affected features for the final results using the proposed method called CFS-RF. Finally, the system uses the proposed algorithms HABBAs as a classifier to detect potential attacks or normal activities.

Figure 2 depicts the detailed structure of the proposed system. It is composed of sequential stages where each stage consists of a number of steps, each of which performs a specific work. The input for the next stage is provided by

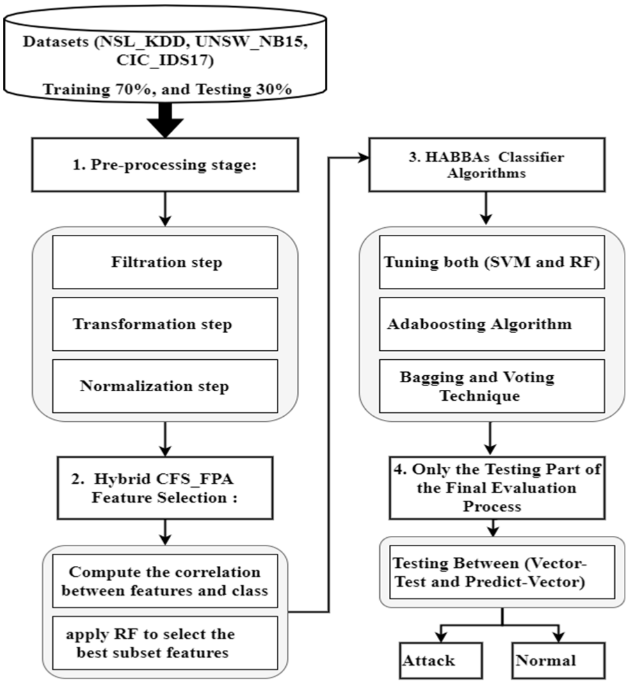

Datasets (NSL_KDD, UNSW_NB15, CIC_IDS17)
Training 70%, and Testing 30%
1. Pre-processing stage:
Filtration step
Transformation step
Normalization step
2. Hybrid CFS_FPA Feature Selection :
Compute the correlation between features and class
apply RF to select the best subset features
3. HABBAs Classifier Algorithms
Tuning both (SVM and RF)
Adaboosting Algorithm
Bagging and Voting Technique
4. Only the Testing Part of the Final Evaluation Process
Testing Between (Vector-Test and Predict-Vector)
Attack
Normal

FIGURE 2. The detailed structure of the anomaly detection system.

the previous stage. These stages and steps are explained in detail successively. The collected datasets (i.e., NSL_KDD, UNSW_NB2015, and CIC_IDS2017) are being read, then perform the preprocessing stage that consists of three main steps: (filtration, transformation, and normalization).

The FS stage selects the best subset of features using the proposed CFS-RF method.

The classifiers' training stage is performed by building the hybrid adaboosting bagging algorithms (HABBAs), modifying the classifiers (RF and SVM) to work as adaboosting, and aggregating the composite model to work as a bagging algorithm. The main reasons behind integrating these two algorithms are the lack of accuracy and susceptibility to model overfitting in the adaboosting and bagging algorithms, respectively. Thus, HABBAs tend to achieve greater accuracy with less overfitting.

The attacks recognition stage is accuracy verification during the testing process of comparing the original and prediction tests using the CFS-RF and HABBAs with the weighting average voting technique.

The classifications evaluation stage applies specific performance measurements (i.e., Accuracy, Recall, Precision, F-measure, DR, and FAR) using two types of each dataset form (i.e., binary and multi-class classifications).

B. PREPROCESSING STAGE

1) DATASETS DESCRIPTION

This system uses three different datasets to implement experiments: NSL_KDD, UNSW_NB2015, and CIC_IDS2017.

The NSL-KDD is the first dataset. It was developed to improve the prediction complications as an influential parameter. Various baseline classifiers were adopted for records categorization of 5 complexity degrees with the number of accurate predictions provided notes next to each occurrence [12]. The percentage of records in the initial KDDCup'99 dataset chosen for every difficulty degree classification is inversely correlated with the number of records selected. KDD_Train set had 125.973 occurrences in our sample, including 58.630 occurrences of attacks and 67.343 of regular traffic. The second dataset (UNSW-NB15) incorporates the bulk of existing low-key attacks in an effort to mimic current network settings. It had 2,540,044 records of 4 big-data CSV files, training/testing records of 175,341 /82,332, and 45 columns (id=1, features=44). Finally, CIC_IDS2017 contains benign data and latest widespread attacks [37] and the results of the CIC flow meter network traffic analysis. The protocols, source/destination IPs, ports, and all attacks were time-stamped flows (CSV files). Moreover, its dataset is most recent, including updated DDoS, Brute Force, XSS, SQL Injection, Infiltration, Port Scan, and Botnet attacks. It had 2,830,743 records of 8 files, whereas every record includes 78 different labeled features.

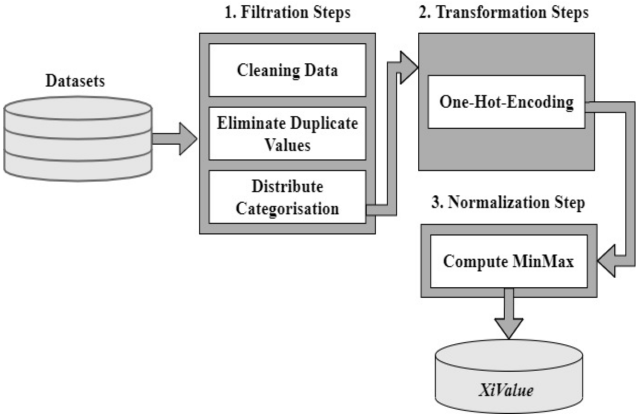


FIGURE 3. Preprocessing stage (first step of the proposed system).

2) PREPROCESSING STEPS

The preprocessing transforms the raw data into an analysis-ready form and then applies it in three steps. These steps (i.e., filtration, transformation, and normalization) are demonstrated in figure 3, whereas algorithm 1 explains the preprocessing stage.

a: THE FILTRATION STEP

It removes unwanted or meaningless data from the datasets, redistributes the resulted data, and rearranges it into categorized groups to make the datasets easier to understand and treat.

b: THE TRANSFORMATION STEP

It converts the non-numerical attribute data into numerical using a one-hot encoding function, which transforms categorical features into numerical values, for instance, converting protocol types (e.g., user datagram protocol (UDP)

Algorithm 1 Preprocessing (Filtration, Transformation, and Normalization)

```
Input: Read Datasets D1, D2, and D3. Where */D1 is NSL, D2 is UNSW_NB2015, and D3 is CIC_IDS2017*/
Output: XiValue(feature)
Begin
1. Loop
     Steps 1, and 2: For data-filtration and data-transformation do
               Delete redundant instances and meaningless ones.
               Distribution categorization is arranged.
          If non-numeric-input then do
               To get numbers: categorical features transform.
               applied (One-Hot Encoding) function.
          End if
        End For
2.   Step 3: Normalization applied Minimax as follows:
          Max =Finding the Maximum value.
          Min =Finding the Minimum value.
          $\text{XiValue} = \frac{\text{XiValue} - \text{Min}}{\text{Max} - \text{Min}}$
3. Stop criteria until all features are done.
4. Return XiValue
End
```

and transmission control protocol (TCP)) into numerical data using this function.

c: THE NORMALIZATION STEP

It applies the Minimax function to convert values between zero and one.

C. HYBRID CFS-RF METHOD

We develop a hybrid strategy for efficient FS and accurate classification by combining CFS and the bagging EL (RF). The system utilizes the proposed hybrid CFS-RF for FS, as explained in algorithm 2 and figure 4. The proposed hybrid CFS-RF method is detailed as follows:

At first, it takes the result from preprocessing stage (X_i Value), then, applies it to each feature using the merit equation, given by:

$$\mu_s = \frac{k\overline{r_{cf}}}{\sqrt{k + k\,(k-1) + \overline{rff}}} \tag{1}$$

where $\overline{r_{c}f}$ is the correlation between feature and class, $\overline{r_{f}f}$ is the correlation between features. The computed correlation (CFS) explains in example 1. Thereafter, it generates subsets of RF by:

$$\{h\,(x, \theta_k)\,, k{=}\ 1, 2 \ldots\} \tag{2}$$

where h is the RF, K is the integer number, θ_k is the theta, and x is the vector.

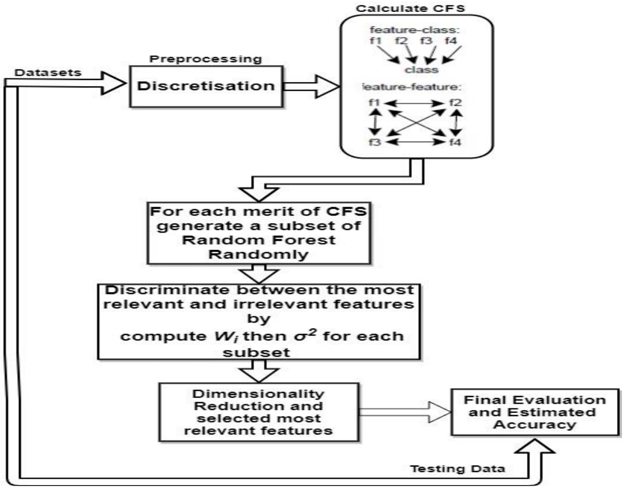

Datasets
Preprocessing
Discretisation
Calculate CFS
feature-class:
f1 f2 f3 f4
class
feature-feature:
f1
f2
f3
f4
For each merit of CFS generate a subset of Random Forest Randomly
Discriminate between the most relevant and irrelevant features by compute W_i then σ^2 for each subset
Dimensionality Reduction and selected most relevant features
Final Evaluation and Estimated Accuracy
Testing Data

FIGURE 4. General structure of the proposed hybrid CFS-RF method.

The process of verifying the redundant features by computing weight range is

$$wR^{\lambda} = \{^{\left[0.000, \frac{-1}{e^{\lambda}}\right], \lambda=1}_{[e^{\frac{-1}{\lambda-1}}+P, e^{\frac{-1}{\lambda}}], \lambda>1} \tag{3}$$

where wR^{λ} is the weight range, and λ is lambda.

It selects the most relevant feature with less variance by computing standard division σ_i, given by:

$$\sigma_i = \frac{1.0{-}\omega_i}{(n+1) - \lambda} \tag{4}$$

where: σ_i is standard division and ω_i is the weight.

Algorithm 2 Hybrid CFS-RF Method

```
Input: Training Datasets after applied algorithm (1), XiValue (feature) // XiValue = features after preprocessing step (algorithm (1)).
Output: Most effective features (XiBest)
Begin
1. For each Xi in the training dataset part Do
      Compute the merit by equation (1).
      Generate 10 RF by equation (2).
   End For
2. XiBest = XiValue
   For i=0 to 10 Do // for each RF generated
   If there are redundant XiBest then
      Compute $wR^{\lambda}$ equation (3). // Compute weight range
      Compute $\sigma_i$ (4).// choose the most relevant F with less variance.
   End if
3. Update $wR^{\lambda}$ and $\sigma^2$
   End For
4. Return the best subset features XiBest
End
```

The proposed algorithm of feature selection (algorithm 2) is explained in example 1 as follows:

TABLE 1. Correlation feature selection.

Feature set	k	$\overline{r_c f}$	$\overline{r_f f}$	μ_s
[]	0	N/A	N/A	0.0
[Du]	1	0.140	1.00	$\frac{1* 0.140}{\sqrt{1+1(1-1)+1.00}}$=0.140
[Proto]	1	0.0210	1.00	$\frac{1* 0.021}{\sqrt{1+1(1-1)+1.00}}$=0.021
[Service]	1	0.0130	1.00	$\frac{1* 0.130}{\sqrt{1+1(1-1)+1.00}}$=0.130
[count]	1	0.185	1.00	$\frac{1* 0.185}{\sqrt{1+1(1-1)+1.00}}$=0.185
[dst_host_ count]	3	0.132	0.0096	$\frac{3* 0.132}{\sqrt{3+3(3-1)+0.0096}}$=0.226
[dst_host_ srv_count]	4	0.105	0.0718	$\frac{1* 0.105}{\sqrt{4+4(4-1)+0.0718}}$=0.192

Example 1: CFS_RF feature selection supposes Probe class of NSL_KDD dataset, explaining the example.

- Training size of the used datasets: NSL_KDD is 125,975 records, UNSW_NB2015 is 175,342 records, and CIC_ID17 is 225746.
- When datasets are applied using algorithm 1: Supposing the NSL_KDD probe class as an example containing 41 features. At first, all the features are called by CFS to compute the correlation between each feature and class using a statistical method. Features of NSL_KDD probe: ([serviceIRC, service_X11, service_Z39, serviceaol, service_auth, service_bgp, service_courier, service_csnet_ns, service_ctf, servicedaytime, service_discard, servicedomain, servicedomain_u, serviceecho, service Eco, service, services exec, servicefinge, serviceftp, serviceftp_data, servicegopher, serviceharvest, servicehostnames, service_http 'service_http74, servicehttp4, servicehttp800, 'service_map4', 'serviceisosap', 'serviceklogin', 'servicekshell', ldapservice, 'serviceink', 'service_login', 'servicemtp', 'servicename', 'servicenetbios', 'service_netbios_ns', 'servicenetbiossn', 'service_netstat', 'servicensp', 'service_ntp', 'servicentp', 'service_other', 'servicepm_dump', 'servicepop2', 'servicepop3', 'serviceprinter', 'service_private', 'serviceredi', 'service_remotejob', 'service_rje', 'service_shell', 'service, 'servicesqlnet', 'servicessh', 'servicesunrpc', 'service_supd', 'servicesystat', 'servicetelnet', 'servicetftp, 'servicetim, 'servicetime', 'serviceurhi', 'serviceurp, 'serviceuucp',serviceuucp, 'servicevmnet', 'servicewhois']).

TABLE 1 computes the CFS using the merit equation, finding CFS for some features to explain the idea.

TABLE 1 demonstrates that less correlation is 0.021 and high correlation is 0.226.

- When applying CFS and obtaining μs in this state, it applies RF with penalizing attributes for these μs randomly using 10 estimators (10 subsets) and ensemble learning (RF). For each set, it computes *Wi* selecting the highest weight and ignoring lower weights. Ultimately, only the best influential subset features will be selected (i.e., the features that affect the intrusion detection performance).

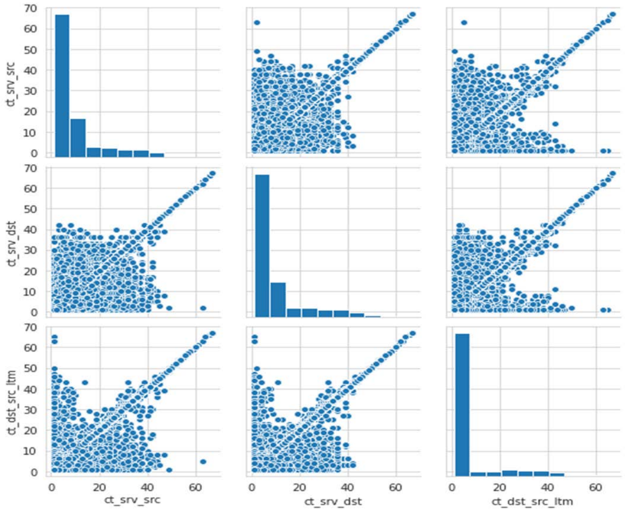


FIGURE 5. Analysis Datasets (compute correlations between features).

Figure 5 explains the analysis dataset before FS.

Figure 5 explains the features' correlation and distribution in the dataset. It is noticeable that the most correlated features are:

- ct_srv_src, ct_dst_src_ltm, ct_srv_dst
  1. All features range between 0 to 60.
  2. Most values are close to 0 and less than 10.
  3. Values are well scattered, however, there is a clear line, indicating some linear relationship.

The hybrid CFS-RF method reduces dimensionality and eliminates superfluous attributes from the dataset.

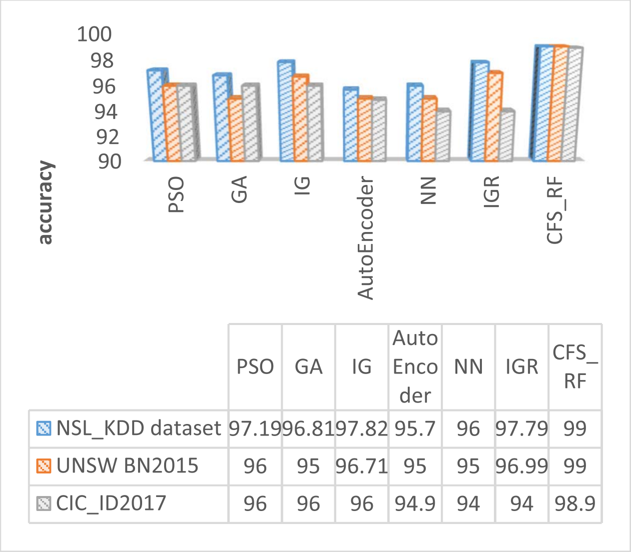
At this end, the analysis and distribution of datasets of the proposed CFS-RF result in 30 features for NSL_KDD, 35 features for UNSW_NB2015, and 40 features for CIC_IDS2017. TABLES 2 summarizes the outputs of the base classifiers using the NSL_KDD, UNSW_NB2015, and CIC_IDS2017 datasets sequentially when applied to the proposed modified HABBAs.

TABLE 2 demonstrates the performance measurement and the execution time in four states (i.e., all, 35, 40, and 30-features). They show that the 30, 35, and 40 features selected are the best in evaluating system measurements (i.e., accuracy, precision, F-measure, and DR are 0.99%). Moreover, they are the best features for reducing execution time to 0.539, 0.839, and 0.931 sec.

By conducting tests on the three datasets, we compare the proposed CFS-RF with several common FS methods, e.g., information gain (IG) [38], IGR information gain ratio (IGR) [39], genetic algorithm (GA) [40], particle swarm optimization (PSO) [41], neural network (NN)[42], and Auto Encoder [43].

TABLE 2. Applied HABBAs for proposed FS using NSL-KDD dataset.

NSL_KDD Performance results using original features (30 features)						
Method	accuracy	Precision	F-measure	DR	FAR	Execution-Time
CFS_RF	**0.997**	**0.998**	**0.995**	**0.998**	**0.004**	**0.539**
UNSW_NB2015 Performance results using CFS-RF (35-features)						
Method	accuracy	Precision	F-measure	DR	FAR	Execution-Time
CFS_RF	**0.990**	**0.99**	**0.992**	**0.990**	**0.008**	**0.839**
CIC_IDS2017 Performance results using CFS-RF (40-features)						
Method	accuracy	Precision	F-measure	DR	FAR	Execution-Time
CFS_RF	**0.99**	**0.98**	**0.99**	**0.99**	**0.001**	**0.931**



	PSO	GA	IG	AutoEncoder	NN	IGR	CFS_RF
▨ NSL_KDD dataset	97.19	96.81	97.82	95.7	96	97.79	99
▨ UNSW BN2015	96	95	96.71	95	95	96.99	99
▨ CIC_ID2017	96	96	96	94.9	94	94	98.9

FIGURE 6. Accuracy measures of the CFS-RF Method.

This comparison research employs standard measures such as Accuracy, F-Measure, DR, and FAR. For the efficiency measurement of the presented IDS, a comparison was made for the number of features and the selection time. Compared with several FS methods, the proposed work outperforms state-of-the-art FS-based approaches on each dataset with accuracy measurement, as demonstrated in figure 6.

Figure 6 explains that all the methods use a different number of FS and use F-Measure to verify the accuracy score of each classifier used in these various datasets with a variety of execution time values as it is high and may take hours when computed. It shows that all results are not convincing, whereas the proposed CFS-RF is the best. This measure depends essentially on two specific parts of recall and precision to verify all the records in the datasets with less time complexity.

D. (HYBRID ADABOOSTING AND BAGGING ALGORITHMS) TRAINING-TESTING AND RECOGNITION ATTACK

Hybrid EL algorithms are built in this stage. The successive classifiers (i.e., RF and SVM) are modified to work sequentially as adaboosting using their updated weights for achieving a convenient performance.

1) MODIFIED RF CLASSIFIER

Algorithms 3 explains the modified RF to work as adaboosting. The parameters and weights are also modified to increase the efficiency of detecting unknown attacks. At first, the initialization process equalizes all values of the XiBest with Wi and generates the RF subsets using equation (5). Afterwords, for each training set, it computes the weight and standard deviation by:

$$\mathrm{p}\sigma^2 + \frac{1-\mathrm{p}}{\mathrm{B}}\sigma 2 \tag{5}$$

where B is a constant and P denotes population.

It is very important to compute the σ for each XiBest in algorithm 3 as a stopping criteria condition.

Then, the proposed model aggregates these classifiers to work in parallel as bagging, using the weighting average voting technique to achieve the best results of these modified classifiers. Algorithm 5 presents the main idea of the proposed HABBAs.

Algorithm 3 Modified RF to work as AdaBoosting

```
Input: XiBest features from each dataset after algorithm 2.
Output: Performance-Measurements.
Initialize:
Begin
  1. XiBest= the best subset of each dataset after executing
     the algorithm 2.
  2. For each Xibest Do
       Assign weight value to each XiBest =Wi. // where
     Wi=0.
  3. End For
  4. Generate new RF using equation (2) //Generating No.
     of RF as 10_forests(estimators).
  5. For a training set XiBest Do /*each generated XiBest
     from RF*/.
       Compute wi
       Compute $\sigma$ for each XiBest by equation (5)
       If $\sigma > 0$ then
         Update Wi for each XiBest
       End If
     End for
  6. Compute measurements: Accuracy, DR, FAR
  7. Return Performance-Measurements
End
```

In algorithm 3, according to weights updates, the RF is modified to work sequentially as adaboosting. In order to achieve the best results of variance and bias, aggregation is performed and applied to other modified classifiers using the weighting average voting technique. This algorithm is modified resulting in a better performance with the least error-rate. The algorithm's general work is depicted in figure 7.

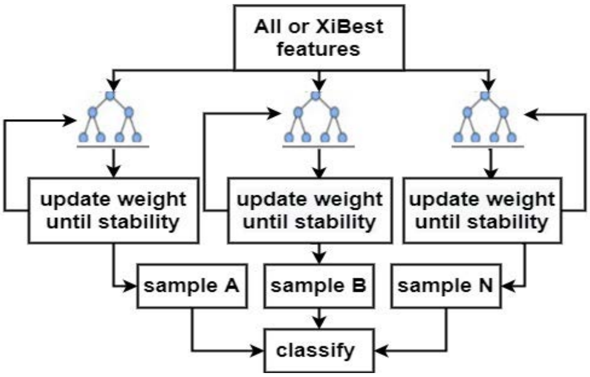

All or XiBest features
update weight until stability
update weight until stability
update weight until stability
sample A
sample B
sample N
classify

FIGURE 7. Block diagram of modified random forest classifier.

2) MODIFIED SVM CLASSIFIER

Algorithm 4 performs the SVM mathematical and computational operations, which requires empirical time. It updates the weight of each XiBest feature in the datasets. Figure 8 depicts the steps of splitting the datasets classes using the modified SVM classifier and computing the support vector using:

$$f(x) = (w, x) + b \tag{6}$$

where: W is the weight and B is the bias.

```
Algorithm 4 Modified SVM to Work as AdaBoosting
Input: XiValue features from each dataset after algorithm 2.
Output: Performance-Measurements.
Begin
1. Initialize: XiBest= XiValue, XiBest = Wi. //where Wi=0.
2. Split classes of each training dataset using hyber_plain into two classes (positive and negative).
3. For each XiBest in the training set Do
       Determine the support vectors using linear.
       Compute F(x) for each support vector using equation (6).
       update the weight for each XiBest.
       Select high Wi
4. End for
5. Compute measurements: Accuracy, DR, FAR
6. Return Performance-Measurements
End
```

Algorithm 5 consists of two main steps; the first step implies applying the adaboosting algorithm for each modified classifier (i.e., algorithms 3 and 4) by computing the weight for each classifier using:

$$(\text{Mi}) = \sum_{j=1}^{d} wj \times err(Xj) \tag{7}$$

where wj is the weight and $err(Xj)$ error for each classifier.

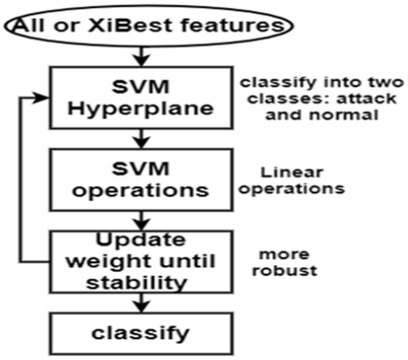


FIGURE 8. Block diagram of modified SVM classifier.

Then, by verifying the error rate by computing the weight using:

$$[\log (1\text{- ErrorRate (Mi))/ ErrorRate (Mi)}] \tag{8}$$

This results in variance reduction and performance enhancement.

The second step applies the principle of bagging algorithm to enable these classifiers to work as bootstrap, then performs aggregation by weighting average voting technique to obtain a composite model with less bias and overfitting (overfitting is reduced in the proposed system by using less depth tree, a sample number of variables during each splitting process, using different dataset) by using:

$$\text{Average voting} = \frac{1}{\text{mj}} = 1 \sum_{\text{i=1}}^{\text{l}} \text{pci}(\frac{\text{wi}}{\text{x}}). \tag{9}$$

Equation 9 computes the voting average technique.

IV. IMPLEMENTATION

The proposed system is implemented using three different datasets (i.e., NSL_KDD, UNSW_NB2015, and CIC_IDS2017). The training dataset used is 70%, whereas the rest 30% is the testing dataset dedicated to evaluating the proposed system. Performance evaluation of the proposed work is examined by executing it with three different selected features, i.e., 13, 30, and all using CFS_RF for the NSL KDD dataset, 13, 35, and all for UNSW_NB2015, and 13, 40, and all for CIC_IDS2017. Afterby, the potential intrusions will be detected using HABBAs and two types of Confusion-Matrices (i.e., binary class and multi-class) classification forms. Finally, system performance is evaluated using different measurements; recall, precision, DR, FAR, and FNR. It is implemented by software (i.e., python 3.8 and colab with sklearn library) using computer hardware with the following technical specifications: Core i7 CPU, 10 TH GEN, and 64-bit OS Windows 11.

V. EXPERIMENTAL RESULTS AND EVALUATION

A. BINARY CONFUSION-MATRIX

HABBAs are implemented using three datasets. The Confusion-Matrix is manually applied to each class, which

Algorithm 5 Hybrid HABBAs for Intrusion Detection

```
Begin
Training part:
1. AdaBoosting-Algorithms:
        Initialization: Wi=0/∗weight of each class∗/;
        k=2/∗(RF, SVM algorithms) ∗/; Ci is the classifiers.
   Looping /∗For each i=1 to modified classifiers (Ki) do∗/
        Appliedequation (7) to compute Mi/∗ Mi means
        Error-Rate ∗/.
   If Mi > 0.5 then/∗ if error rate is larger than 0.5 ∗/
        Equation (8) is applied to compute for each
        algorithm (Wi).
        Computing: Ci=Mi /∗ each modified classifier
        Mi compute prediction∗/
        For every two classifiers, Ci (i.e., RF, and SVM)
        adds Wi.
    End if
   Until the two classifiers, Ci is done
   Return Minimum Mi,
2. Bagging-Algorithms:
   Looping:/∗For modified classifiers Ci (i.e., SVM, RF) ∗/
        Applied the principle of the ensemble to each Ci /∗
        using
        model of bootstrap∗/.
        Equation (9) is used to apply the principle of voting.
   Until the two classifiers, Ci is done
   Testing part: (do the following steps):
   Looping
        After voting computes, the accuracy for each
        (prediction) after and before to achieve both
        Xi-After, and If Xi-Before.
   If Xi-Before is greater than Xi-After, then
        Replacing process:( voting average with the highest
        probability).
   else
        Accuracy, FAR, FNR, DR, and FAR are computed.
   End if
   Until the stopping criterion is done.
3. Return Measurements, and A composite model.
End
```

has both normal and abnormal traffics. The FSs (i.e., 13, 30, 35, 40, and all features) are applied to the presented CFS-RF and HABBAs to detect intrusions. The proposal applies a Confusion-Matrix in binary class form.

The proposal is applied to the NSL_KDD classes. TABLE 5 shows the application of NSL_KDD with 30 FS as a binary class. This table shows the 4 states' classification, i.e., True-Positive (TP), False-Positive (FP), True-Negative (TN), and False-Negative (FN). System performance measurements depend on these four states. The tables show that the confusion matrices explain the number of attacks and normal distribution of each class, indicating that the best results are achieved when applying 30-features. TABLE 3 depicts the accuracy and FNR of all these tables.

TABLE 3. NSL_KDD confusion-MATRIX FOR 30-features.

	Prediction	
	Attack	Benign
Attack	9714	1
Benign	0	2885

TABLE 4. UNSW_NB2015 dataset Confusion-MATRIX FOR 35-features.

	Prediction	
	Attack	Attack
Attack	*1701*	*25*
Benign	*0*	*744*

TABLE 5. CIC_IDS2017 dataset confusion-MATRIX FOR 40-features.

	Prediction	
	Attack	Attack
Attack	453916	349
Benign	369	110928

TABLE 3 explains the binary class of NSL_KDD, showing that the 30-features subset is the best FS.

The main purpose of using various datasets is to discover new attacks, making the system more robust against external and new attacks (zero-day attacks). TABLE 6 explains the accuracy and FNR of all these features in detail. It addresses that the best results of accuracy and FNR can be achieved when applying 30-features to the proposed system. FNR is the division of false-negative detections divided by false-negative and true-positive detections in an experiment. This measurement is considerably important to evaluate the performance and quality of the proposed system by computing the number of errors discovered for each attack diagnosed as normal. In addition, when applying 13 and 41 features, the FNR and accuracy measures are insufficient comparably.

TABLES 6 depicts the binary class of UNSW_NB2015 with 35 features, whereas TABLE 8 shows TP, TN, FP, and FN with accuracy measures for all the selected features in these tables.

TABLE 6 explains the binary class of the UNSW_NB2015, showing that the best FS to be applied is 35-features due to the correct distribution and diagnosis of the attacks and normal. The details explained in TABLE 6 show a higher accuracy of 99% when applying 35-features with a low FNR of 0.01.

CIC_IDS2017 is the final dataset to test the proposed system in the same number of FSs. TABLE 5 shows the Confusion-Matrix of CIC_IDS2017 with 40 features as a binary class. TABLE 6 depicts TP, TN, FP, and FN with accuracy measures for all FSs and datasets.

TABLE 6 shows the highest accuracy of 0.99% with an FNR of 0.0008 when applying the proposed system with 40-features. The lowest accuracy of applying 13-features is 0.87% with an FNR of 0.123, while applying 78-features achieves an accuracy of 0.92% with an FNR of 0.053.

TABLE 6. Accuracy applied HABBAs by different FSs.

Datasets	Features number	TP	FN	Accuracy	FNR
NSL_KDD	13	9000	605	9000+2280/12600 =0.89	605/(605+9000) =0.06
	30	9714	0	9714+2885/12600 =0.99	0/(0+9714)= 0
	41	9500	405	9500+2480/12600 =0.95	405/(405+9500) =0.04
UNSW_NB2015	13	1500	344	1500+400/2470=0.76	344/344+1500=0.19
	35	1701	25	1701+744/2470=0.99	25/25+1701=0.01
	49	1525	201	1525+630/2470=0.87	201/201+1525=0.11
CIC_IDS2017	13	443615	62736	492176/565562=0.87	62736/443615+62736=0.123
	40	453916	369	564844/565562=0.99	369/369+453916 =0.0008
	78	437550	24741	524106/565562=0.92	24741/24741+437550=0.053

TABLE 7. Attacks accuracy and f-SCORE WHEN applying the proposed system.

id	Type of Attack	Accuracy	F-score
0	DDoS	98%	99%
1	Port Scan	100%	99%
2	Brute Force	98.4%	100%
3	DoS Hulk	99%	99%
4	DoS GoldenEye	98.9%	99%
5	DoS Hulk	100%	99%
6	Bot	100%	100%
7	DoS slow loris	99%	99%
8	FTP Patator	98%	99%
9	SSH-Patator	100%	99%
10	XSS	100%	100%
11	Benign	99%	99%

Table 7 depicts the accuracy of each attack in the dataset with the f-score measure.

Table 7 addresses all classes' best results when applying the proposed system, reaching 100% in XXS and Bot. It indicates that the number of features is ideal and helpful for identifying all forms of attacks.

B. THE COMPLEXITY TIME AND RUN TIME

It includes the computations of complexity time for the presented work by computing Big-O notation, which is O (N^2). However, figures 9, 10, and 11 explain the running time applying NSL_KDD, UNSW_NB2015, and CIC_IDS2017, respectively, showing the highest and lowest values. In figure 9, the highest is 9.6 sec. in the DoS class and the lowest is 1.3 sec. in the R2L class. While in figure 10, the highest is 4.1 sec. in Fuzzers class and the lowest is 0.01 sec. in the Shellcode class. Finally, in figure 11, the highest is 11.5 sec. in DDoS_ston class, while the lowest is 1.1 sec. in the brute force class. Hence, the running time increases when the input increases; thus, it is proportional to the number of inputs.

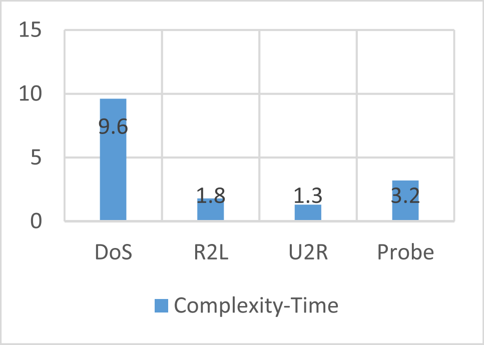


FIGURE 9. NSL_KDD dataset complexity time.

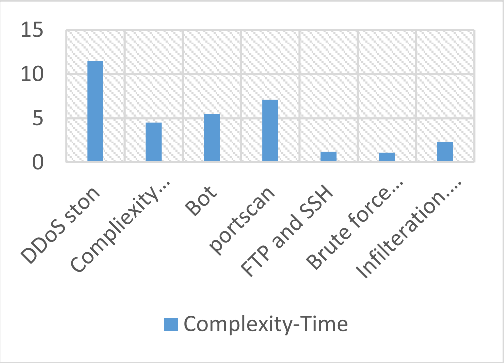


FIGURE 10. UNSW_NB2015 dataset complexity-time.

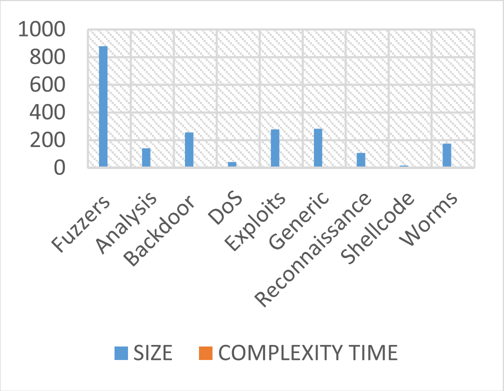


FIGURE 11. CIC_IDS2017 dataset complexity-time.

C. COMPARISON WITH OTHER STUDIES

The proposed system is examined and compared to other studies in terms of accuracy, FAR, DR, number of FS, FS method, and classification method. The detection accuracy of the proposal is 99% for training and 90% for testing. it yields a higher DR with a lower FAR comparing to the single-stage approach. This trade-off is elaborated in

TABLE 8. Results comparison of the proposed system with other studies.

References with years	Dataset	FS method	Classification method	Number of FS	Accuracy %	DR %	FAR %
[25], 2016	NSL_KDD	Decision Tree	Rule-based Ensemble Learning	N/A	80	81	N/A
[26], 2017	NSL_KDD	KNN	Naive Bayes	16	83	82	4.83
[27], 2021	NSL_KDD	IG, symmetrical uncertainty, and correlation-based feature subset	GAR Forest	32	85	N/A	15.00
				10	78	N/A	1.00
[28], 2021	NSL_KDD	Entropy	SVM	42	95	96	5.11
[29], 2021	NSL_KDD	Wrapper based GA	Logistic regression as ensemble algorithm	13	97.99	96.64	N/A
	UNSW_NB2015			8	98.73	98.93	N/A
	CIC_ID17			11	98.99	98.75	N/A
[30], 2022	NSL_KDD		Gradient Boosting algorithm	N/A	99	N/A	N/A
[30], 2022	CIC_ID2017	Deep NN		N/A	92	N/A	N/A
[33], 2022	CIC_ID2017	Panalized attribute	(NB,RF,KNN,SVM)		87	99	0.04
[34], 2022	NSL_KDD	correlation-based feature	N/A	30	85	N/A	N/A
	UNSW_NB2015	selection					
Proposed system	NSL_KDD	CFS-RF	Voting (RF, and SVM)	30	99.4	99.9	0.004
	UNSW_NB2015			35	99.8	99.6	0.008
	CIC_ID17			40	99.7	99.4	0.0012

TABLE 8. Furthermore, in comparison with previous studies, it achieves the best accuracy, DR, and FAR throughout the entire testing process.

D. ANALYSIS RESULTS

Preprocessing stage is essential to prepare the dataset for the feature selection stage (CFS-RF). In the CFS-RF stage, each class in the dataset undergoes an analysis process to verify and select only the best influential subset features that affect the final results. Ultimately, CFS-RF selects the most appropriate features subset of the datasets (i.e., 30 features of NSL_KD, 35-features of the UNSW_NB201, and 40 features of CIC_IDS201. Afterby, the classifiers stage starts to make each classifier (SVM and RF) work as adaboosting (sequentially) and aggregates using the voting average technique to work as a bagging algorithm (in parallel).

E. LIMITATIONS

The main objective of the proposed work is to distinguish between normal and abnormal activities to increase system robustness against new attacks. However, it has the following limitations:

The HABBAs system achieves an excellent performance when applying dataset attacks but does not take into account more attacks launched by external networks (when available).

In the HABBAs system training phase, once it is completed, the values of the training part are fixed (i.e., 70%), making it difficult to migrate to detect more attacks.

VI. CONCLUSION AND FUTURE WORK

Despite adopting various ML strategies previously to improve IDSs' efficacy, the existing IDSs are still unsuccessful, significantly to cope with the vulnerability of the expected wireless paradigms. Based on the preferred hybrid methods in FS and EL algorithms, this paper developed a unique IDS method to adapt to the imbalanced and high-dimensional traffic with low DR. A hybrid CFS-RF method was presented to achieve the optimal subset of function correlation using 30-features for NSL_KDD, 35-features for UNSW_NB2015, and 40-features for CIC_IDS2017 sample with a hybrid EL method. The results showed an accuracy of 0.99% for all the datasets, while FAR values were 0.004, 0.008, and 0.0012 for the NSL_KDD, UNSW_NB2015, and CIC_IDS2017 datasets, respectively. Hence, other parametric values are detailed in the results comparison table. Moreover, the proposed method outperformed the existing classification algorithms. As demonstrated, this method provided a significant competitive edge to the IDS market compared to other strategies. Despite the privilege of CFS-RF with ensemble algorithms (HABBAs), more extensive work is still required to expand system capacity to treat infrequent traffic hazards in the future. The authors recommend analyzing a stream of data connections can help detect the undetectable assaults by applying IDS to each connection record individually and employing the proposed NIDS on the systems' confidential servers of security establishments. Apparently, the proposed system is considerably an excellent and robust system for detecting intrusions on the network rapidly, providing high accuracy.

REFERENCES

[1] H. W. Oleiwi and H. Al-Raweshidy, "Cooperative SWIPT THz-NOMA/6G performance analysis," *Electronics*, vol. 11, no. 6, p. 873, Mar. 2022, doi: 10.3390/electronics11060873.

[2] H. W. Oleiwi, N. Saeed, and H. Al-Raweshidy, "Cooperative SWIPT MIMO-NOMA for reliable THz 6G communications," *Network*, vol. 2, no. 2, pp. 257–269, Apr. 2022, doi: 10.3390/network2020017.

[3] X. Sun, J. Dai, P. Liu, and A. Singhal, and J. Yen, "Using Bayesian networks for probabilistic identification of zero-day attack paths," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 10, pp. 2506–2521, Oct. 2018, doi: 10.1109/TIFS.2018.2821095.

[4] M. Alazab, "Profiling and classifying the behavior of malicious codes," *J. Syst. Softw.*, vol. 100, pp. 91–102, Feb. 2015, doi: 10.1016/j.jss.2014.10.031.

[5] I. S. Thaseen and C. A. Kumar, ''Intrusion detection model using fusion of chi-square feature selection and multi class SVM,'' *J. King Saud Univ. Comput. Inf. Sci.*, vol. 29, no. 4, pp. 462–472, Oct. 2017, doi: 10.1016/j.jksuci.2015.12.004.

[6] W. Liang, L. Xiao, K. Zhang, M. Tang, D. He, and K.-C. Li, ''Data fusion approach for collaborative anomaly intrusion detection in blockchain-based systems,'' *IEEE Internet Things J.*, vol. 9, no. 16, pp. 14741–14751, Aug. 2022, doi: 10.1109/JIOT.2021.3053842.

[7] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, ''CorrAUC: A malicious bot-IoT traffic detection method in IoT network using machine-learning techniques,'' *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3242–3254, Mar. 2021, doi: 10.1109/JIOT.2020.3002255.

[8] M. A. Khan and J. Kim, ''Toward developing efficient Conv-AE-based intrusion detection system using heterogeneous dataset,'' *Electronics*, vol. 9, no. 11, pp. 1–17, 2020, doi: 10.3390/electronics9111771.

[9] M. Safaldin, M. Otair, and L. Abualigah, ''Improved binary gray wolf optimizer and SVM for intrusion detection system in wireless sensor networks,'' *J. Ambient Intell. Hum. Comput.*, vol. 12, no. 2, pp. 1559–1576, Feb. 2021, doi: 10.1007/s12652-020-02228-z.

[10] A. R. Javed, S. U. Rehman, M. U. Khan, M. Alazab, and T. Reddy, ''CANintelliIDS: Detecting in-vehicle intrusion attacks on a controller area network using CNN and attention-based GRU,'' *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 1456–1466, Apr. 2021, doi: 10.1109/TNSE.2021.3059881.

[11] S. Rajagopal, P. P. Kundapur, and K. S. Hareesha, ''A stacking ensemble for network intrusion detection using heterogeneous datasets,'' *Secur. Commun. Netw.*, vol. 2020, pp. 1–9, Jan. 2020, doi: 10.1155/2020/4586875.

[12] S. Mokhtari, A. Abbaspour, K. K. Yen, and A. Sargolzaei, ''A machine learning approach for anomaly detection in industrial control systems based on measurement data,'' *Electronics*, vol. 10, no. 4, pp. 1–13, 2021, doi: 10.3390/electronics10040407.

[13] T. Sommestad, H. Holm, and D. Steinvall, ''Variables influencing the effectiveness of signature-based network intrusion detection systems,'' *Inf. Secur. J., Global Perspective*, vol. 2021, pp. 1–18, Sep. 2021, doi: 10.1080/19393555.2021.1975853.

[14] E. A. Winanto, M. Y. Idris, D. Stiawan, and M. S. Nurfatih, ''Designing consensus algorithm for collaborative signature-based intrusion detection system,'' *Indones J. Electron. Eng. Comput. Sci.*, vol. 22, no. 1, pp. 485–496, 2021, doi: 10.11591/ijeecs.v22.i1.pp485-496.

[15] G. Creech and J. Hu, ''A semantic approach to host-based intrusion detection systems using contiguousand discontiguous system call patterns,'' *IEEE Trans. Comput.*, vol. 63, no. 4, pp. 807–819, Apr. 2014, doi: 10.1109/TC.2013.13.

[16] K. K. Sahu, S. C. Nayak, and H. S. Behera, ''Multi-step-ahead exchange rate forecasting for South Asian countries using multi-verse optimized multiplicative functional link neural networks,'' *Karbala Int. J. Modern Sci.*, vol. 7, no. 1, p. 7, Mar. 2021, doi: 10.33640/2405-609X.2278.

[17] M. Jabardi and A. S. Hadi, ''Twitter fake account detection and classification using ontological engineering and semantic Web rule language,'' *Karbala Int. J. Modern Sci.*, vol. 6, no. 4, pp. 404–413, Dec. 2020, doi: 10.33640/2405-609X.2285.

[18] V. A. Dovgal and D. V. Dovgal, ''Security analysis of a swarm of drones resisting attacks by intruders,'' in *Proc. CEUR Workshop*, vol. 2914, 2021, pp. 316–323.

[19] L. Vinet and A. Zhedanov, ''A 'missing' family of classical orthogonal polynomials,'' *J. Phys. A, Math Theory*, vol. 44, no. 8, pp. 1–14, 2011, doi: 10.1088/1751-8113/44/8/085201.

[20] H. S. Hota and A. K. Shrivas, ''Decision tree techniques applied on NSL-KDD data and its comparison with various feature selection techniques,'' in *Smart Innovation Systems and Technologies*, vol. 27, no. 1. Cham, Switzerland: Springer, 2014, pp. 205–212, doi: 10.1007/978-3-319-07353-8_24.

[21] C. Khammassi and S. Krichen, ''A GA-LR wrapper approach for feature selection in network intrusion detection,'' *Comput. Secur.*, vol. 70, pp. 255–277, Sep. 2017, doi: 10.1016/j.cose.2017.06.005.

[22] S.-H. Moon and Y.-H. Kim, ''An improved forecast of precipitation type using correlation-based feature selection and multinomial logistic regression,'' *Atmos. Res.*, vol. 240, Aug. 2020, Art. no. 104928, doi: 10.1016/j.atmosres.2020.104928.

[23] M. Mohamad, A. Selamat, O. Krejcar, R. G. Crespo, E. Herrera-Viedma, and H. Fujita, ''Enhancing big data feature selection using a hybrid correlation-based feature selection,'' *Electronics*, vol. 10, no. 23, p. 2984, Nov. 2021, doi: 10.3390/electronics10232984.

[24] M. Loey, G. Manogaran, M. H. N. Taha, and N. E. M. Khalifa, ''A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic,'' *Measurement*, vol. 167, Jan. 2021, Art. no. 108288, doi: 10.1016/j.measurement.2020.108288.

[25] D. Gaikwad and R. Thool, ''DAREnsemble: Decision tree and rule learner based ensemble for network intrusion detection system,'' in *Smart Innovation Systems and Technologies*, vol. 50. Cham, Switzerland: Springer, 2016, doi: 10.1007/978-3-319-30933-0_20.

[26] H. H. Pajouh, G. Dastghaibyfard, and S. Hashemi, ''Two-tier network anomaly detection model: A machine learning approach,'' *J. Intell. Inf. Syst.*, vol. 48, no. 1, pp. 61–74, Feb. 2017, doi: 10.1007/s10844-015-0388-x.

[27] N. K. Kanakarajan and K. Muniasamy, ''Improving the accuracy of intrusion detection using gar-forest with feature selection,'' in *Advances in Intelligent Systems and Computing*, vol. 404. New Delhi, India: Springer, 2016, pp. 539–547, doi: 10.1007/978-81-322-2695-6_45.

[28] M. Mittal, R. P. de Prado, Y. Kawai, S. Nakajima, and J. E. Muñoz-Expósito, ''Machine learning techniques for energy efficiency and anomaly detection in hybrid wireless sensor networks,'' *Energies*, vol. 14, no. 11, p. 3125, May 2021, doi: 10.3390/en14113125.

[29] E. Jaw and X. Wang, ''Feature selection and ensemble-based intrusion detection system: An efficient and comprehensive approach,'' *Symmetry*, vol. 13, no. 10, p. 1764, Sep. 2021, doi: 10.3390/sym13101764.

[30] N. Gupta, V. Jindal, and P. Bedi, ''CSE-IDS: Using cost-sensitive deep learning and ensemble algorithms to handle class imbalance in network-based intrusion detection systems,'' *Comput. Secur.*, vol. 112, Jan. 2022, Art. no. 102499, doi: 10.1016/j.cose.2021.102499.

[31] S. N. Mighan and M. Kahani, ''A novel scalable intrusion detection system based on deep learning,'' *Int. J. Inf. Secur.*, vol. 20, no. 3, pp. 387–403, Jun. 2021, doi: 10.1007/s10207-020-00508-5.

[32] C. A. de Souza, C. B. Westphall, R. B. Machado, J. B. M. Sobral, and G. D. S. Vieira, ''Hybrid approach to intrusion detection in fog-based IoT environments,'' *Comput. Netw.*, vol. 180, Oct. 2020, Art. no. 107417, doi: 10.1016/j.comnet.2020.107417.

[33] D. N. Mhawi, A. Aldallal, and S. Hassan, ''Advanced feature-selection-based hybrid ensemble learning algorithms for network intrusion detection systems,'' *Symmetry*, vol. 14, no. 7, p. 1461, Jul. 2022, doi: 10.3390/sym14071461.

[34] D. N. Mhawi, ''Proposed hybrid correlation feature selection forest panalized attribute approach to advance IDSs,'' *Karbala Int. J. Modren Sci.*, vol. 7, no. 4, p. 15, 2021.

[35] M. A. Khan, M. R. Karim, and Y. Kim, ''A two-stage big data analytics framework with real world applications using spark machine learning and long short-term memory network,'' *Symmetry*, vol. 10, no. 10, p. 485, Oct. 2018, doi: 10.3390/sym10100485.

[36] M. Kulariya, P. Saraf, R. Ranjan, and G. P. Gupta, ''Performance analysis of network intrusion detection schemes using apache spark,'' in *Proc. Int. Conf. Commun. Signal Process. (ICCSP)*, Apr. 2016, pp. 1973–1977, doi: 10.1109/ICCSP.2016.7754517.

[37] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, ''Toward generating a new intrusion detection dataset and intrusion traffic characterization,'' in *Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy*, 2018, pp. 108–116, doi: 10.5220/0006639801080116.

[38] A. I. Pratiwi and Adiwijaya, ''On the feature selection and classification based on information gain for document sentiment analysis,'' *Appl. Comput. Intell. Soft Comput.*, vol. 2018, pp. 1–5, Feb. 2018, doi: 10.1155/2018/1407817.

[39] L. Li, X. Zhang, and M. Xue, ''Explaining information gain and information gain ratio in information theory,'' *ICIC Exp. Lett*, vol. 7, no. 8, pp. 2385–2391, 2013.

[40] L. M. Patnaik and S. Mandavilli, ''Adaptation in genetic algorithms,'' in *Genetic Algorithms for Pattern Recognition*. Boca Raton, FL, USA: CRC Press, 2017, pp. 45–64, doi: 10.1201/9780203713402.

[41] Y. Zhang, S. Wang, and G. Ji, ''A comprehensive survey on particle swarm optimization algorithm and its applications,'' *Math. Problems Eng.*, vol. 2015, pp. 1–38, Feb. 2015, doi: 10.1155/2015/931256.

[42] S. T. Ikram, A. K. Cherukuri, B. Poorva, P. S. Ushasree, Y. Zhang, X. Liu, and G. Li, ''Anomaly detection using XGBoost ensemble of deep neural network models,'' *Cybern. Inf. Technol.*, vol. 21, no. 3, pp. 175–188, Sep. 2021, doi: 10.2478/cait-2021-0037.

[43] T. Poongothai, K. Jayarajan, and S. Martin, ''An effective and intelligent intrusion detection system using deep auto-encoders professor,'' *Paideuma J.*, vol. 13, no. 5, pp. 39–52, 2020.