symmetry

MDPI

Article

Advanced Feature-Selection-Based Hybrid Ensemble Learning Algorithms for Network Intrusion Detection Systems

Doaa N. Mhawi [1], Ammar Aldallal [2,*] and Soukeana Hassan [3]

[1] Computer Science Department, University of Technology, and Middle Technical University, Baghdad 10010, Iraq; dododuaaenteesha@mtu.edu.iq
[2] Telecommunication Engineering Department, Ahlia University, Manama P.O. Box 10878, Bahrain
[3] Computer Science Department, University of Technology, Baghdad 10010, Iraq; soukaena.hassan@yahoo.com
* Correspondence: aaldallal@ahlia.edu.bh

Abstract: As cyber-attacks become remarkably sophisticated, effective Intrusion Detection Systems (IDSs) are needed to monitor computer resources and to provide alerts regarding unusual or suspicious behavior. Despite using several machine learning (ML) and data mining methods to achieve high effectiveness, these systems have not proven ideal. Current intrusion detection algorithms suffer from high dimensionality, redundancy, meaningless data, high error rate, false alarm rate, and false-negative rate. This paper proposes a novel Ensemble Learning (EL) algorithm-based network IDS model. The efficient feature selection is attained via a hybrid of Correlation Feature Selection coupled with Forest Panelized Attributes (CFS–FPA). The improved intrusion detection involves exploiting AdaBoosting and bagging ensemble learning algorithms to modify four classifiers: Support Vector Machine, Random Forest, Naïve Bayes, and K-Nearest Neighbor. These four enhanced classifiers have been applied first as AdaBoosting and then as bagging, using the aggregation technique through the voting average technique. To provide better benchmarking, both binary and multi-class classification forms are used to evaluate the model. The experimental results of applying the model to CICIDS2017 dataset achieved promising results of 99.7%accuracy, a 0.053 false-negative rate, and a 0.004 false alarm rate. This system will be effective for information technology-based organizations, as it is expected to provide a high level of symmetry between information security and detection of attacks and malicious intrusion.

Keywords: correlation feature selection; Cybersecurity; ensemble learning; Forest Panelized Attribute; intrusion detection system; machine learning

Citation: Mhawi, D.N.; Aldallal, A.; Hassan, S. Advanced Feature-Selection-Based Hybrid Ensemble Learning Algorithms for Network Intrusion Detection Systems. *Symmetry* **2022**, *14*, 1461. https://doi.org/10.3390/sym14071461

Academic Editor: Chien-Hsing Chou

Received: 25 June 2022
Accepted: 12 July 2022
Published: 17 July 2022

1. Introduction

Every day, different types of new cyber-attacks are discovered, and their sources are becoming more hazardous. As a result, detecting zero-day attacks is a difficult operation that potentially jeopardizes business continuity [1]. Computer attacks are becoming increasingly complex, posing difficulties in accurately detecting the intrusion [2,3]. Network Intrusion detection systems (NIDSs) are meant to monitor computer networks for unusual activities that a regular packet filter would miss Traditional IDSs have a number of flaws, such as the inability to discriminate between new malicious threats; the need for modification; poor accuracy; and a high rate of false alerts. Therefore, machine learning is used to detect new attacks. However, machine learning encounters many challenges because it enhances the computational and time complexity of the task by expanding the search space [4,5]. Numerous studies have been conducted on the use of multiple classifiers instead of single ones and the principle of ensemble learning techniques to ensure high accuracy and a low false alarm rate [6–8]. As a result, ensemble learning can be divided into three categories (i.e., bagging, stacking, and boosting) [9–11]. It is a general meta approach to machine learning to combine predictions from multiple models to improve predictive performance. Although

an infinite number of ensembles for any predictive modeling can be created, the subject of ensemble learning is dominated by three methods. The first category of the ensemble is bagging. It is the process of fitting multiple decision trees to different samples of the same dataset and then averaging the results [12]. Alternatively, the stacking method involves fitting many different model types onto the same data and using another model to learn how to best combine the predictions [13]. Boosting involves sequentially adding ensemble members that correct the predictions made by prior models and output a weighted average of the predictions [14]. Recently, researchers applied hybrid principles in feature selection and ensemble methods. Feature selection is a useful approach for intrusion detection systems. This method discovers extremely important features and discards unnecessary ones while causing minimal performance reduction [15–17]. Correlation-based feature selection (CFS) selects strong affinity on similarities that are used as a heuristic evaluation function. The function compares feature vector subsets that are related to the class label but are not associated with one another. The CFS algorithm implies that irrelevant characteristics have a low association with the class and should be removed consequently. Excess traits, on the other hand, should be investigated since they are typically associated with one or more of the other characteristics [18]. The classification algorithms used in ensemble learning frequently mix numerous basic classifiers in some fashion. The proposed work will benefit from this feature of ensemble learning by developing and integrating multiple distinct models, these classifiers are effective at dealing with the same problem and, when combined, produce a predicting output that is more stable and accurate. To begin with, a single classifier may not always be competent to produce the best representation in the hypothesis space. Thus, the use of multiple independent classifiers is necessary to improve prediction performance. Second, a false or inaccurate hypothesis can develop if the training dataset for the learning algorithm is insufficient.

This paper proposes a dimensionality reduction approach as well as the Feature Selection (FS) method for obtaining the optimum subset of the original features. The IDS's stability and accuracy are then improved by submitting these subsets to the proposed hybrid ensemble learning, which requires minimum computational and time resources. The present study is significant because it aims to:

- Reduce the dimensionality of the CICIDS2017 dataset through the proposed coupling of Correlation Feature Selection with Forest Panelized Attributes.
- Find the best machine learning (ensemble method) approach to collect the four modified classifiers (Support Vector Machine, Random Forest, Naïve Bayes, and K-Nearest Neighbor) to ensure the best result of the hybrid ensemble method.
- Conduct a comparative study between the CFS–FPA and other features selection techniques in terms of accuracy, Detection Rate (DR), and False Alarm Rate (FAR). The outcome will be used to generalize the efficiency of the proposed features selection technique.
- Compare the four classifiers before and after modification and work as the AdaBoosting method. In addition, comparing the proposed method with other existing approaches.

The remainder of the paper is organized as follows: Section 2 includes a review of similar feature selection techniques and ML-based IDS. Section 3 defines the suggested system, approach, and distinct proposed ML. Section 4 presents the experimental data, discussion, and findings. Finally, Section 5 presents the conclusion and future work.

2. Related Work

Recently, researchers focused on developing ML-based IDS using two well-known datasets: NSL-KDD, and CICIDS2017. Zhou et al. [19] proposed an IDS based on feature selection and ensemble classifier. This framework is based on feature selection and ensemble learning techniques. In the first step, both heuristic algorithm CFS and Bat Algorithm (BA) are proposed for dimensionality reduction. In the second step, an ensemble approach that combines C4.5 and Random Forest (RF) algorithms is applied. Finally, it performs a voting technique using NSL-KDD, AWID, and CICIDS2017 datasets. The experimental results of

this work reach 84% accuracy in the testing and a 0.15 false alarm rate; 96% accuracy and a 94% detection rate with 10 selected features when applied to NSL KDD datasets; and 94.5% and 92% for accuracy and detection rates, respectively, for UNSW BN15 dataset with 13 features.

Jaw and Wang [20] proposed a Comprehensive Approach for IDS. A wrapper methodology based on a genetic algorithm is adopted as a feature selection and logistic regression as an ensemble learning algorithm for network intrusion detection systems. Experimental results show excellent performance accuracy of 98.99%, 98.73%, and 97.997%, and detection rates of 98.75%, 96.64%, and 98.93% for CICIDS2017, NSL-KDD, and UNSW-NB15, respectively, based on only 11, 8, and 13 selected relevant features from the above datasets.

Gupta et al. [21] recommended that ensemble algorithms handle a class imbalance in network-based intrusion detection systems. This work consisted of three stages. The first stage is the deep neural network for splitting and discriminating normal from suspicious traffic network attacks and then for classifying major attacks using the eXtreme Gradient Boosting algorithm as the second stage. The final stage uses Random Forest to classify the minor attacks. This model used NSL-KDD, CIDDS-001, and CICIDS2017 datasets to evaluate the performance of the proposed system. The accuracy achieved was 99% for NSL, 96% for CIDDS-001%, and 92% for CICIDS2017, and complexity time was measured in hours, not in minutes.

Tama et al. [22] used a hybrid feature selection method with two stages of ensemble learning classifiers. CIC-IDS2017 dataset with 37 features was used to evaluate the performance of the proposed system, and the accuracy was 96.46%.

The IDS proposed by Aldallal and Alisa [23] merges genetic algorithm (GA) and support vector machine (SVM), where GA is used to select an optimal set of features from the CICIDS2017 dataset, while SVM is applied to classify the network traffic into benign and abnormal. The results obtained by using CICIDS2017 outperform those obtained when using KDD CUP 99 and NSL-KDD by up to 5.74%.

Pelletier and Abualkibash, in [24], proposed a model to detect intrusions on the network by applying Neural Network as a feature selection method and Random Forest algorithm as a classifier to detect the intrusion. This model is tested by using CIC-IDS2017 dataset, and the experimental result of the accuracy reached 97.30% whereas the number of features used in this model was 30 features.

Abbas et al. in [25] proposed a new ensemble-based intrusion detection system for the Internet of Things. Those researchers used different deployed classifiers (i.e., logistic regression, naive Bayes, and a decision tree) with voting technique. The experimental result using CICIDS2017 with two forms (binary, and multi-class).

An architectural model is presented in [26] for risk assessment (RA) of the information system with the CICIDS2017 dataset using ML algorithms. ML techniques including K nearest neighbors (KNN), NB, gradient boosting tree, RF, and decision tree (DT) were evaluated for RA in this study. The performance of the model was based on the ML technique that has efficient predictively of intrusion. The predictive model was the implementation of ML techniques that produced better results with the CICIDS2017 dataset. For RA, the risk matrix was analyzed by 15 models with predicted results.

All previous works suffer from conflict in the measured values, where some of them are sufficient in the accuracy but not sufficient in other measures, such as [19] where accuracy reached 96% while FAR is 0.15%, [20] where accuracy reached 98.3% while FAR is 0.14%. Hence, the proposed system increases robustness by using an advanced feature selection method based on hybrid ensemble learning algorithms aiming at achieving high accuracy and minimal FAR.

3. Materials and Methods

The proposed system provides an efficient ML-based IDS that uses new hybrid FS ensemble learning techniques with a voting classifier that is a group of classifiers. It is proposed to enhance the detection capabilities of IDS to protect service providers against

attacks. Figure 1 depicts the block diagram of the main idea of the proposed Hybrid AdaBoosting and Bagging Algorithms (HABBAs).

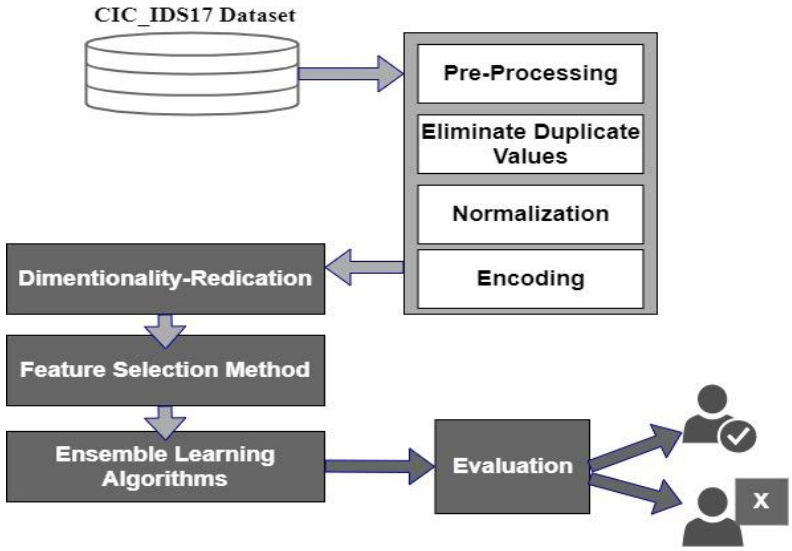

Figure 1. General block diagram of the proposed system.

Figure 1 consists of several stages starting from collecting the data and ending with detecting normal or attack traffic. The following subsections provide an informative explanation of the framework.

3.1. Description of CICIDS2017 Datasets

It is a challenging effort for researchers to find an appropriate dataset for evaluating IDSs. This paper applied the CICIDS2017 dataset for experiments. The Canadian Institute for Cybersecurity (CIC) issued the CIC IDS2017 dataset in 2017. It includes benign data and the most recent common attacks [13]. The results of the CIC flow meter network traffic analysis are also included. Protocols, source and destination IPs, ports, and attacks all have time-stamped flows. This dataset is one of the most updated datasets. It includes updated DDoS, Brute Force, XSS, SQL Injection, Infiltration, Port Scan, and Botnet assaults. This dataset has 2,830,743 records which are spread across eight files. Each record contains 78 various features with their labels. The Wednesday-working hours' set is chosen for experimentation using the cross-validation method to retain the same magnitude order of each dataset when multi-classification is needed. Table 1 shows the statistical information for this set, which contains 691,406 occurrences divided into six categories.

Table 1. CICIDS2017 dataset.

Classes	CIC_IDS/Wen.
DoS-slow loris	5499
DoS-Slow-HTTPtest	5796
DoSHulk	10,293
DoS-Golden-Eye	230,124
Hear-bleed	11
Normal	439,683
Total	691,406
Attack	251,723

3.2. CICIDS17 Dataset Preprocessing

Figure 2 depicts the steps of this stage in detail. Processes at this stage are done on the CICIDS17 dataset that is formatted as CSV. In this stage, raw data is transformed into an

analysis-ready format. This stage consists of three steps. These steps are: (1) filtration, when data is cleaned and duplicate values are removed; (2) transformation, when Label-Encoder and One-Hot Encoding techniques are applied; and (3) normalization, when minimax function is used to scale values between zero and one. The algorithm of this stage is explained in Algorithm 1.

Algorithm 1: Preprocessing and Minimax Scaling

Input: Read d1 where d1 is CICIDS201
Output: Normalize the dataset to d1normalize.
Begin
For each Di dataset Do
Step 1: Data Filtering
Removed meaningless and redundant instances.
Arrange Distribution-categorization.
Step 2: Data transformation
if (do non-numeric input) then do:
Transform categorical features into numbers using:
Label Encoder ()
One-Hot Encoding /*this process is a complement to the categorical transform that is used to convert categorical features into numbers such as convert protocol types such as UDP, and TCP into numerical data using this function) */
End if
Step 3: Normalization
Minimax scaling is computed by applying the following:
Max = Find the Maximum value.
Min = Find the Minimum value.
For each XiValue in the dataset Do
$$\text{XiValue} = \frac{\text{XiValue} - \text{Min}}{\text{Max} - \text{Min}}$$
Return XiValue Between [0, 1]
Remove missing and duplicated data
Encoding process with the second normalization
End For
End For
End

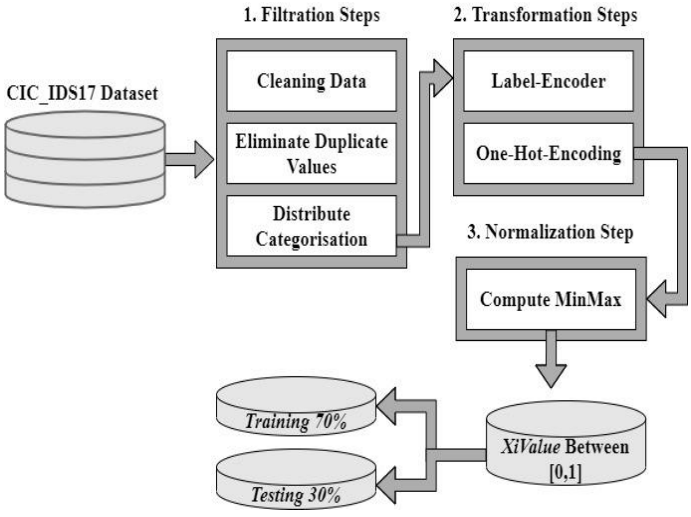

Figure 2. Preprocessing stage.

3.3. Correlation Feature Selection-Forest Panelized Attribute (CFS-FPA)

This proposed method is explained in detail in [27,28]. It is used to reduce dimensionality and select the best subset features. Based on this method, the best 30 features are selected out of 78 features of the CICIDS17 dataset. Table 2 depicts these 30 features. Figure 3 depicts the main steps of the proposed FS.

Table 2. Subset feature selection of CICIDS2017.

CICIDS2017
Port-Destination
Flow-Duration
FlowIATStd
FlowIATMax
Flow_IAT_Min
Fwd_IAT_Total
Fwd_IAT_Std
Fwd_IAT_Max
Fwd_IAT_Min
BwdIATStd
BwdIATMax
BwdIATMin
FwdPSHFlags
MaxPacketLength
PacketLengthMean
PacketLengthStd
PacketLengthVariance
FINFlagCount
SYNFlagCount
PSHFlagCount
ACKFlagCount
IdleMean
'IdleMax
IdleMin
Destination_Port
Flow_Duration
PSHFlagCountID
Bwd_Packet_Length_Max
Bwd_Packet_Length_Max
BwdIATStd

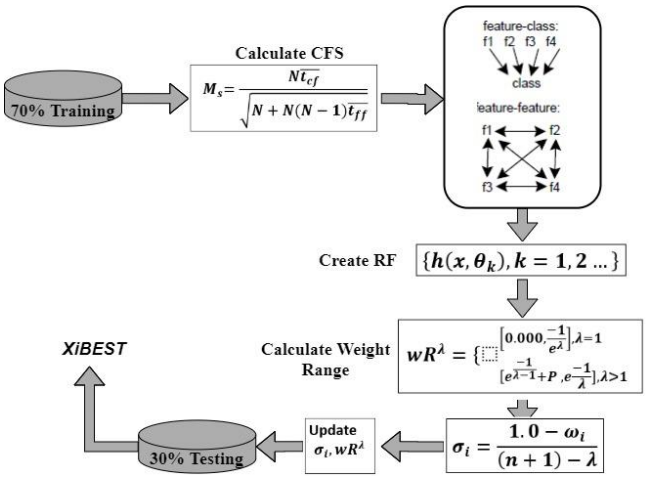

Figure 3. CFS-FPA stage.

In Figure 3, Correlation Feature Selection combined with Forest Panelised Attributes (CFS–FPA) is used to analyze the correlation of the selected features and is effective for enhancing the efficiency of the training and testing phases.

3.4. Classifiers

The IDS proposed in this work is based on four classifiers. A brief explanation of these classifiers is presented here:

3.4.1. Random Forest (RF)

Sekulić suggested Random Forest in [29]. This is a decision tree methodology that works by constructing many decision trees. It categorizes hundreds of input variables based on their importance without eliminating any one of them. RF is a set of classification trees, each of which devotes a single vote to the task of identifying the most common class in the input data. SVM and ANN, for example, have smaller parameters when RF is used instead of other machine learning algorithms. In RF, a set of tree-structured classifiers can be defined as follows:

$$\{h(x, \theta k), \quad k = 1, 2, 3, \dots \} \tag{1}$$

In this model, h denotes an RF classifier and k is a collection of identical vectors dispersed at random.

Each tree has a vote for the most renowned class at input variable x. Its utilization has an impact on the proportions and design of the tree structure. Establishing each decision-making tree is critical to RF's success.

In RF, which has a minimal calculation cost, outliers and parameters have little impact. Furthermore, compared to a single DT, overfitting is less of an issue, and the trees do not need to be pruned [30]. With a volatility of two, the variance of an average of Bagging random variables has a 1/B2 volatility. The average variance is then computed using Equation (2), and if it is more than zero, the weight Wi for each subset feature is updated (XiBest).

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

Here, σ^2 is a stander division, p is population, and B is a constant.

3.4.2. Naïve Bayes Classifier

Naïve Bayes is one of the most widely used classifiers that is based on statistical classification. It is a form of supervised ML algorithm. It is featured by surprisingly usefulness and high accuracy due to possessing several properties. It is characterized by a strong independence probabilistic classifier. In which, for a given class variable, the presence or absence of a feature is unrelated to the absence or presence of another feature. In a supervised learning setting, Naïve Bayes classifiers can be trained very efficiently depending on the precise nature of the probability model, [31]. Basically, it has two variables:

Class variable (C), and a set of attributes F = $\{A_1; A_2; \dots ; A_n\}$, on a dataset D which consists of instances $\{I_1, I_2, \dots , I_t\}$ and can be defined as in Equation (3), assuming the that the attributes are independent within the class as in Equation (4). Figure 4 demonstrates the structure of Naïve Bayes [32], where C is the classifier and A1, A2, ... are the attributes.

$$\mathrm{c(E)} = \underset{c \in C}{\operatorname{argmax}} \; P(c) \times P(a_1 \, , 3_2 \, , \; \dots , \; a_n | c) \tag{3}$$

$$\mathrm{P(E|c)} = P(a_{\,1} \, , a_2 \, , \; \dots , \; a_n | c) = \prod_{i=1}^{n} P(a_i | c) \tag{4}$$

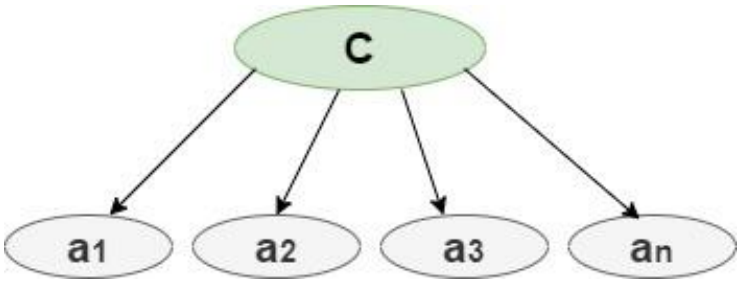

Figure 4. NB Classifier.

The conditional independence assumption leads to posterior probabilities. The NB classifier is constructed easily because of the simplicity of computing *P(C)* and $P(a_i|c)$. It simplifies computations and provides high accuracy and speed when applied to large databases.

3.4.3. Support Vector Machine (SVM)

A statistical classifier is the use of a single-class was suggested by [33,34]. It is possible to predict the support of a high-dimensional distribution. It uses relaxation parameters to isolate the test point of a class from the rest of the datasets after first processing features with a kernel. Iterative relaxation parameter methods are used to solve massive sparse linear systems. It is also used to solve problems involving linear least-squares and nonlinear equations.

The classifier converts instances into a large dimensional attribute space (via a kernel) and finds the best boundary hyperplane position to break the training data according to the following formula [31]:

$$f(x) = (w, x) + b \quad (5)$$

where w refers to the normal vector and b refers to a bias term.

By optimizing rule f, SVM changes the hyperplane to find a linear classifier. A mark can be assigned to a test example x using this classification law. If the result of $f(x)$ is less than zero, x is classified as an intrusion; otherwise, it is classified as natural. As presented in Figure 5, the classification situation can be clarified by the product of $f(x)$: Positive is classified as regular, and negative is classified as an intrusion.

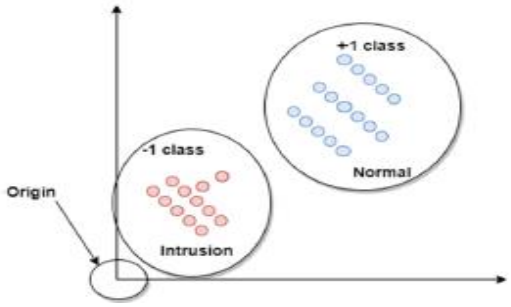

Figure 5. SVM classifier based on relaxation parameter [31].

3.4.4. K-Nearest Neighbor (KNN)

According to the distance function, the nearest neighbor classifier (NNC) assigns a class to the given test pattern that is the same as its nearest neighbor in the training set. The k-nearest neighbor classifier (k-NNC) is a generalization of NNC, where k is an integer and k = 1. The training set contains k-nearest neighbors for the given test pattern Y. Each of the k closest neighbors' class information is maintained. In most circumstances, NNC using bootstrap samples outperforms traditional k-NNC according to experiments. It is worth noting that there is no theoretical explanation for why k-NNC, which uses the bootstrapped dataset, is better [35]. Figure 6 depicts KNN.

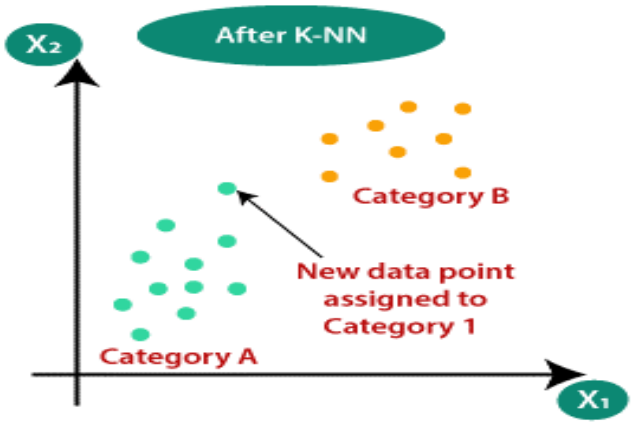

Figure 6. KNN classifier [35].

3.5. Hybrid Classifier Algorithms

Hybrid ensemble learning algorithms are built during this stage. At first, four different classifiers (i.e., SVM, RF, NB, and KNN) are used to facilitate sequential operation. The weight is updated for an effective performance using the principle of AdaBoosting. These classifiers are modified to work as AdaBoosting to run sequentially after modifying the weight in order to achieve a high weight with less variance and bias to produce better results when aggregated and applied with other modified classifiers by using the voting technique (Overfitting is avoided through reducing tree depth, number of samples of variables at each split and using different dataset). Therefore, these algorithms are modified in this manner to obtain better results and performance with a minimal error rate. Figures 7–10 demonstrate block diagrams for SVM, RF, NB, and KNN, respectively. Algorithm 2 depicts the proposed Hybrid AdaBoosting and Bagging Algorithms (HABBAs).

Figure 7, CF classifier, considers the best subset features (XiBEST) after applying preprocessing and CFS_FPA [28]. Thereafter it initializes the weight (*Wi*), and creates the subset forest by using Equation (1).

Figure 8 depicts the block diagram of SVM Classifier. It starts with the initialization step to set the weight *Wi* to zero and to begin the splitting process for the training dataset using a hyperplane. Next, it uses Equation (5) explained earlier to compute the function that utilizes each of *Wi*, bias, and vector of training data. The modified SVM overcomes two main drawbacks of classical SVM, the first one is that it is not suitable for large databases, and the second one is that it does not perform very well when the dataset has more noise. As the support vector classifier works by putting data points, above and below the classifying hyperplane, there is no probabilistic explanation for the classification.

Figure 9 depicts the process of computing the probability of each subset feature (XiBEST) and finding the maximum values to update the weight of each XiBEST.

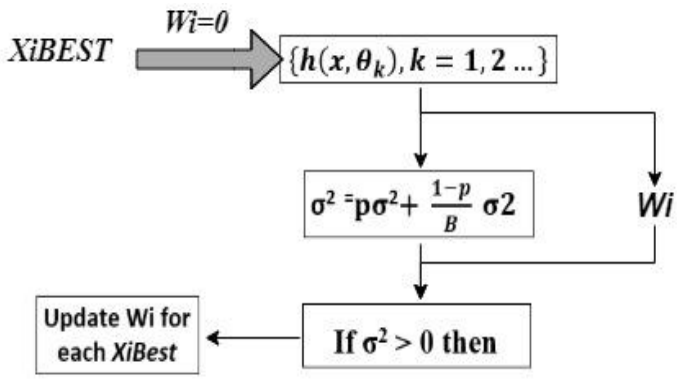

Figure 7. Block diagram of RF Classifier.

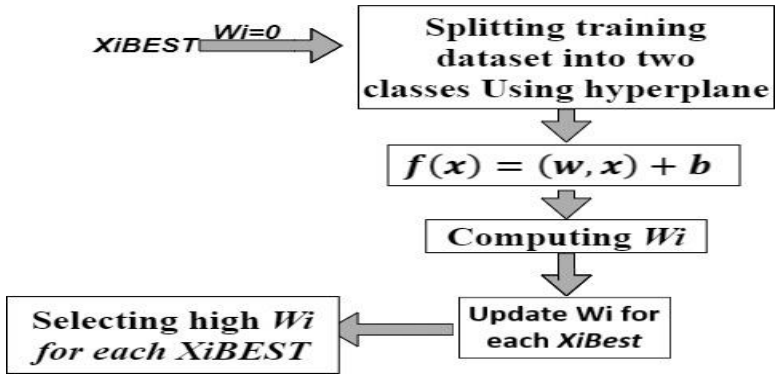

Figure 8. Block diagram of SVM Classifier.

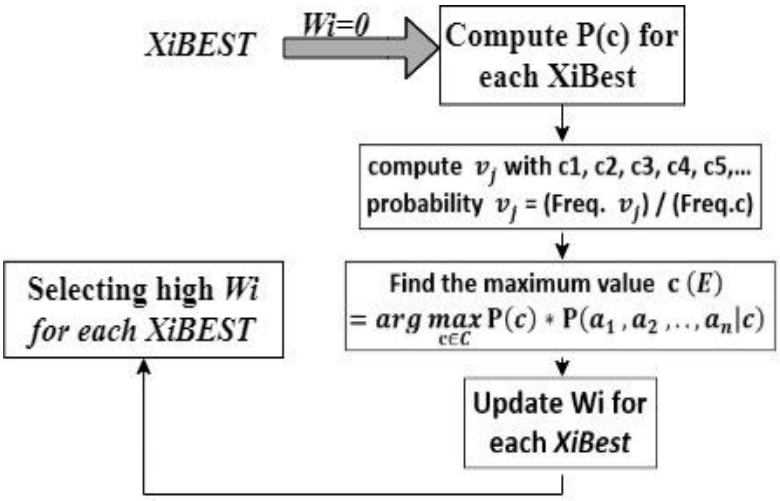

Figure 9. Block diagram for NB Classifier.

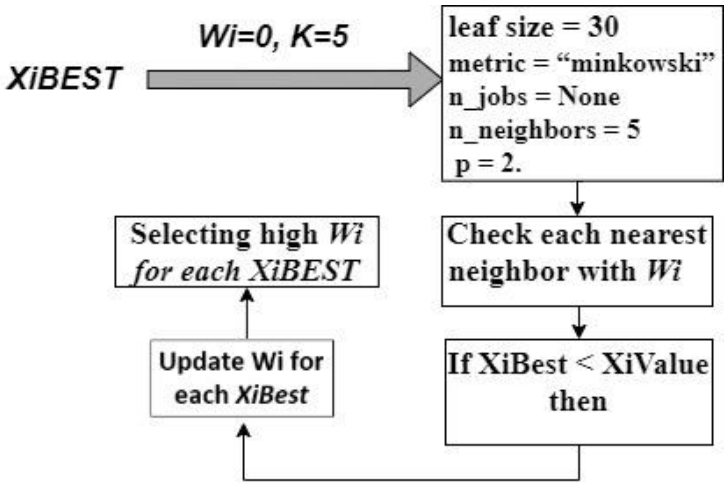

Figure 10. Block diagram for KNN Classifier.

Algorithm 2: Hybrid HABBAs for Intrusion Detection

Input: D1 = CICIDS17 training datasets; Mi = modified classifiers, k = the number of rounds (one Modified algorithm per round);
Output: A composite model
Step 1: Adaboosting Algorithms
initialize weight of each class Wi = 0; // this weight for each modified algorithm.
k = 4; (four modified algorithms).
for i = 1 to k do // for each modified algorithm
Compute ErrorRate (Mi)
If ErrorRate (Mi) > 0.5 then
compute Wi to each k.
[log (1 − ErrorRate (Mi))/ErrorRate (Mi)]
compute prediction of each modified classifier Mi: Ci = Mi(x)
add Wi to weight for classifier Ci
End if
End for
Return Ci with the highest weight and error rate
Step 2: bagging Algorithms
For each Ci Do
Ensemble these Ci to bootstrap models.
Aggregate each Ci using voting average as parallel operations.
Average voting $\leftarrow \frac{1}{mj} = l \sum_{i=1}^{l} pci(Wix)$.
End for
For the testing set part do:
Compute accuracy for predicted XiAfter voting.
Compute accuracy for predicted XiBefore voting.
If XiBefore < XiAfter then
Return to the Average voting step and replace the probability of weighting using the highest probabilities.
Else
Compute general measurement: Accuracy, DR, FAR, Precision, False-positive Rate, False-negative Rate, True Positive Rate, True negative Rate
End if
End for
Return composite model and Performance-Measurements
End

4. Implementation

This paper aims at building IDSs with better reliability, high accuracy, low false alarm rates, and low false negative rates. CFS–FPA is a proposed method that combines both CFS and FPA. This method applies correlation between features and a target, then it distributes them into subsets using Random Forest. Finally, it uses a panelized attribute to select only features that affect the final results [28]. It enables selecting the best set of features for removing unnecessary features and increasing classification performance with HABBAs to detect intrusion. This proposed system is implemented using the CICIDS2017 dataset to test binary and multi-class forms of the confusion matrix. It is executed using laptop CORE i7, 10th generation with RAM 16. Operated by win11 and Colab platform. Several packages of Sklearn from Python 8.3 are utilized in this model, such as cross_val_score from selection, and Voting Classifier from Ensemble.

5. Experimental Results and Discussion

To evaluate the proposed Modified Ensemble Learning Algorithms, 70% of the dataset is used for training and 30% is used for testing. Testing is done in two stages: feature selection and ensemble algorithms. Table 3 explains the experimental results of using

thirteen different numbers of feature selection along with the accuracy obtained. It is obvious from this experiment that the best accuracy of 99% is achieved when the number of features is 30. Hence it will be adopted for the remaining experiments.

Table 3. Experimental result of 13 different numbers of FS.

Number of FS	Accuracy %
13	74
20	78
25	83
30	99
35	98.9
40	98.5
45	98
50	96.9
55	96.3
60	96
65	94
70	93
78	90

5.1. Binary and Multi-Class Confusion Matrix

The experiment is carried out at this stage. HABBAs use CICIDS2017 dataset by applying a confusion matrix for each class that contains both normal and abnormal traffic and with three sets of feature selections (i.e., 13, 30, and all features). After the applied proposed CFS-FPA method and Hybrid AdaBoosting, a bagging ensemble algorithm is used to detect intrusion.

The confusion matrix is presented in binary and multi-class. At first, the proposed model is applied to each class of CICIDS2017. Tables 4–6 reveal the predicted results when applying the CICIDS2017 dataset with the three feature selection sets (i.e., 13, 30, and 78) as a binary class. Each of these tables explains the distribution of these four states: TP True Positive, FP False Positive, TN True Negative, and FN False Negative, which are used in the calculation of the evaluation measures.

Table 4. Binary CICIDS2017 with 13-features.

ActualClass	PredictedClass	
	Positive	Negative
Positive	443,615	10,650
Negative	62,736	48,561

Table 5. Binary CICIDS2017 with 30-features.

ActualClass	PredictedClass	
	Positive	Negative
Positive	453,916	349
Negative	369	110,928

Table 6. Binary CICIDS2017 with 78-features.

ActualClass	PredictedClass	
	Positive	Negative
Positive	437,550	16,715
Negative	24,741	86,556

The numbers of attacks and normal distribution of each class where the best is when applying 30 features. Table 7 shows the accuracy and FNR of these tables.

Table 7. CICIDS2017 Binary Accuracy with 78-features.

Features	Accuracy	FNR
13	0.87	0.123
30	0.99	0.0008
78	0.92	0.053

Based on the binary confusion matrix, Table 7 shows the highest and the best accuracy obtained by the proposed system, with 30 features chosen using the proposed CFS-FPA method. The lowest accuracy occurs when applied to 13 feature selections. Similarly, it shows the highest accuracy of 99% when applied to 30 selected features and the lowest FNR of 0.0008. This system performs better compared with using 13-features since the accuracy is 87% and FNR is 0.123, and when applying 78-features the accuracy is 0.92% and FNR is 0.053.

Table 8 describes the CICIDS17 confusion matrix of multi-class when applied to 30 features and Table 9 depicts Precision, Recall, and F-score for the same feature selection.

Table 8. Confusion matrix for the CICID2017 dataset for 30 features.

ActualClass	PredictedClass											
	Normal	Bot	Brute Force	DDoS	DoS Golden Eye	DoS Hulk	DoS Slow HTTP Test	DoS Slow Loris	FTP Pastor	Port Scan	Pastor	XSS
Normal	453,761	50	0	0	2	274	3	0	1	174	0	0
Bot	0	391	0	0	0	0	0	0	0	0	0	0
Brute Force	0	0	299	0	0	0	0	0	0	0	0	0
DDoS	10	0	0	25,595	0	0	0	0	0	0	0	0
DoS GoldenEye	8	0	0	0	2042	6	2	0	0	0	0	0
DoS Hulk	21	0	1	2	1	45,999	0	0	0	1	0	0
DoS Slow-HTTP-test	7	0	0	0	0	0	1091	2	0	0	0	0
DoS slow loris	3	0	1	0	0	0	6	1149	0	0	0	0
FTP-Patator	3	0	0	0	0	0	0	0	1584	0	0	0
Port-Scan	1	0	3	0	0	4	0	0	0	31,752	0	1
SSH-Patator	6	0	0	0	0	0	0	0	0	0	1174	0
XSS	0	0	0	0	0	0	0	0	0	0	0	130

Table 9. Accuracy of Multi-class CICIDS2017 with 30-features.

Attack	Precision	Detection-Rate	F-Score
Normal	99	99	100
bot	100	100	100
Brute force	100	100	100
Port-Scan	99	99	99
DoS-slow loris	99	99	99
DoS-Slow-HTTPtest	99	99	99
DoSHulk	99	99	99
DoSGolden-Eye	99	99	99
Hear-bleed	99	99	99
FTP-Patter	99	99	99
SSH-Scan	100	99	99

Table 9 shows that the best results for all classes are achieved when selecting the 30 features reaching 100% in Bot and Brute Force, which means that the features' number is optimal and useful to detect all types of attacks.

5.2. Time Complexity

The time complexity of the proposed algorithm using Big O notation is O(N2) [28]. This means the run-time increases polynomial when the input is increased. The complexity time when applied to the CICIDS17 dataset is presented in Figure 11. It shows that the highest runtime is 11.5 s for DDoS_ston class, while the shortest runtime is 1.1 s for brute force class.

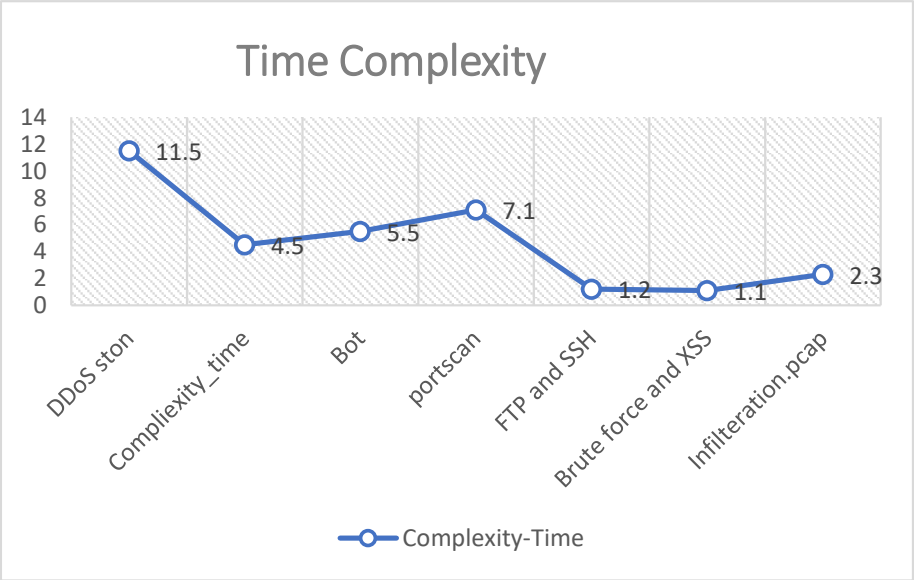

Figure 11. CICID2017 Complexity-time.

5.3. Analysis of Results

To demonstrate the effectiveness of the proposed HABBAs, a comparison study is conducted with similar work. Table 10 reveals that the proposed HABBAs outperform all the selected algorithms for this evaluation. For example, the work done by [30] is applied to the same dataset CICIDS2017, and examined the performance using 10 and 13 features. The best accuracy of 98.4% was when using 10 features compared with 30 in this work. While FAR of 13 features is lower than that of 10 features but still higher than that of our proposed model. Nevertheless, the HABBAs achieved 99.7% accuracy with an improvement of 1.62% and FAR as low as 0.004 [31] applied voting on four ML techniques using three sets of features (8, 11, and 13). These techniques are K-means, SVM, DBSCAN, and Maximization-Expectation. The best average accuracy achieved was 98% which is lower than our model by 1.73%. In the same manner, the best average detection rate is higher than the detection rate of the proposed model. Reference [32] tried three values for the number of features (38, 41, and 78) and the best accuracy achieved is for 41 features. It is 99%, which is slightly lower than that of our model. This study does not provide values for the detection rate of FAR. The accuracy of the work done by [33] is less than the proposed model by 3.4%. The FAR measure of [32] is much higher than the proposed model although the accuracy is 98.5. These results reveal that there is an actual need to have a system that combines high accuracy and low FAR. This system is achieved by the proposed model. Finally, both works of [32] and [33] do not consider the number of features. However, the accuracy of [33] is 97.5% when using the same dataset and it is lower than our model by 2.26%. From the above discussion, it is obvious that the proposed HABBAs model outperforms all the selected algorithms. This is due to the effective feature selection algorithm that obtained

the best combination and most important features which influence the accuracy of the classification of network traffic. This is from one side. On the other side, the proposed voting model of the modified ML algorithms (Support Vector Machine, Random Forest, Naïve Bayes, and K-Nearest Neighbor) demonstrates its ability to accurately classify the network traffic into benign and normal.

Table 10. Comparison with other studies.

References, Authors	Feature Selection Method	Classification Method	FS	Accuracy	DR	FAR
Zhou Y., et al. [19]	CFS_BA	Voting contain (C4.5, RF, ForestPA).	10	98.4	99.1	0.15
			13	97.3	98	0.12
Jaw E. and Wang X. [20]	HFS-KODE	Voting contains (K-means, SVM, DBSCAN, and Maximization-Expectation, (KODE))	11	96.4	99	1.15
			8	98.3	99	0.14
			13	98	98	1.12
Gupta N., et al. [21]	deep neural network	eXtreme Gradient Boosting algorithm	41	99%	__	__
			38	96%	__	__
			78	92%	__	__
Tama B., et al. [22]	hybrid	Ensemble Two-stage	37	96.42	__	__
Pelletier, Z.; Abualkibash, M. [24]	NN	RF	30	97.30%	98%	__
Thaseen I. S. and Ahmad A. [36]	Chi-square	Voting (SVM, MNB, Boosting)	__	98.5	95	2.15
Ikram S. T., et al. [37]	DNN	XGBoost	__	97.5	97	__
Proposed system	**CFS_FPA**	**Voting (RF, NB, KNN, SVM)**	**30**	**99.7**	**99.99**	**0.004**

6. Conclusions

Despite previous attempts to increase the efficacy of IDSs through the use of various ML methods, existing IDSs are still ineffective by some measures. With hybrid techniques based on the desired FS, we proposed a novel IDS approach for dealing with unbalanced and high-dimensional traffic with low DR. A hybrid CFS FPA strategy with a 30-feature sample and a hybrid ensemble learning method is proposed to attain the best subset in terms of function correlation. Removing non-essential features and selecting only affected features through the proposed method CFS_FPA by combining correlation feature selection and forest penalized attribute enabled the proposed system to manipulate and process the conflict that the previous work suffers from such as (FAR, FNR, DR, and accuracy); hence, the accuracy in the testing phase enhanced to 87% and FNR is 0.123. Using the CICIDS2017 dataset, the suggested model's final experimental results showed an accuracy of 99.73%, an F-measure of 99.71%, a precision of 99.82%, a DR of 99.8%, and a FAR of 0.004. Furthermore, the suggested technique outperforms the currently available classification algorithms as well as the previously proposed CFS–FPA–ensemble method. Comparisons with other strategies reveal that this methodology can give a considerable competitive advantage in the IDS industry. Hence, provide high reliability and more robustness in classifying benign traffics and identifying intrusions.

Despite CFS-FPA's advantage with ensemble techniques (HABBAs), additional work is needed to enhance its capabilities to tackle infrequent traffic problems. In the future, we are interested in testing the performance of our proposed model on other datasets such as CICIDS2018 which includes more recent types of attacks. Other types of machine learning techniques could be considered for further enhancement that considers both memory utilization and time complexity to make it more efficient for real-time detection of intrusions and attacks.

Author Contributions: Conceptualization, D.N.M. and S.H.; Formal analysis, D.N.M. and A.A.; Software, D.N.M.; Supervision, S.H.; Writing-original draft, D.N.M.; Writing-review & editing, A.A. and D.N.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: CICIDS2017 Dataset free downloaded from the link: http://205.174.165.80/CICDataset/CIC-IDS-2017/Dataset/, accessed on 24 June 2022.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sun, X.; Dai, J.; Liu, P.; Singhal, A.; Yen, J. Using Bayesian Networks for Probabilistic Identification of Zero-Day Attack Paths. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 2506–2521. [CrossRef]
2. Alazab, M. Profiling and classifying the behavior of malicious codes. *J. Syst. Softw.* **2015**, *100*, 91–102. [CrossRef]
3. Sumaiya Thaseen, I.; Aswani Kumar, C. Intrusion detection model using fusion of chi-square feature selection and multi class SVM. *J. King Saud Univ.—Comput. Inf. Sci.* **2017**, *29*, 462–472. [CrossRef]
4. Rajagopal, S.; Kundapur, P.P.; Hareesha, K.S. A Stacking Ensemble for Network Intrusion Detection Using Heterogeneous Datasets. *Secur. Commun. Netw.* **2020**, *2020*, 4586875. [CrossRef]
5. Aljawarneh, S.; Aldwairi, M.; Yassein, M.B. Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. *J. Comput. Sci.* **2018**, *25*, 152–160. [CrossRef]
6. Sharma, S.; Challa, R.K.; Kumar, R. An ensemble-based supervised machine learning framework for android ransomware detection. *Int. Arab J. Inf. Technol.* **2021**, *18*, 422–429. [CrossRef]
7. Devarajan, R.; Rao, P. An efficient intrusion detection system by using behaviour profiling and statistical approach model. *Int. Arab J. Inf. Technol.* **2021**, *18*, 114–124. [CrossRef]
8. Hnaif, A.; Jaber, K.; Alia, M.; Daghbosheh, M. Parallel scalable approximate matching algorithm for network intrusion detection systems. *Int. Arab J. Inf. Technol.* **2021**, *18*, 77–84. [CrossRef]
9. Aljanabi, M.; Ismail, M. Improved intrusion detection algorithm based on TLBO and GA algorithms. *Int. Arab J. Inf. Technol.* **2021**, *18*, 170–179. [CrossRef]
10. Tabash, M.; Allah, M.A.; Tawfik, B. Intrusion detection model using naive bayes and deep learning technique. *Int. Arab J. Inf. Technol.* **2020**, *17*, 215–224. [CrossRef]
11. Wang, K.; Wang, Y.; Zhao, Q.; Meng, D.; Liao, X.; Xu, Z. SPLBoost: An Improved Robust Boosting Algorithm Based on Self-Paced Learning. *IEEE Trans. Cybern.* **2021**, *51*, 1556–1570. [CrossRef] [PubMed]
12. Wang, C.; Du, J.; Fan, X. High-dimensional correlation matrix estimation for general continuous data with Bagging technique. *Mach. Learn.* **2022**. [CrossRef]
13. Guo, H.W.; Hu, Z.; Liu, Z.B.; Tian, J.G. Stacking of 2D Materials. *Adv. Funct. Mater.* **2021**, *31*, 2007810. [CrossRef]
14. Bentéjac, C.; Csörgő, A.; Martínez-Muñoz, G. A comparative analysis of gradient boosting algorithms. *Artif. Intell. Rev.* **2021**, *54*, 1937–1967. [CrossRef]
15. Hota, H.S.; Shrivas, A.K. Decision tree techniques applied on NSL-KDD data and its comparison with various feature selection techniques. In *Advanced Computing, Networking and Informatics*; Springer: Cham, Switzerland, 2014; Volume 1, pp. 205–212.
16. Khammassi, C.; Krichen, S. A GA-LR wrapper approach for feature selection in network intrusion detection. *Comput. Secur.* **2017**, *70*, 255–277. [CrossRef]
17. Moon, S.H.; Kim, Y.H. An improved forecast of precipitation type using correlation-based feature selection and multinomial logistic regression. *Atmos. Res.* **2020**, *240*, 104928. [CrossRef]
18. Mohamad, M.; Selamat, A.; Krejcar, O.; Crespo, R.G.; Herrera-Viedma, E.; Fujita, H. Enhancing big data feature selection using a hybrid correlation-based feature selection. *Electronics* **2021**, *10*, 2984. [CrossRef]
19. Zhou, Y.; Cheng, G.; Jiang, S.; Dai, M. Building an efficient intrusion detection system based on feature selection and ensemble classifier. *Comput. Netw.* **2020**, *174*, 107274. [CrossRef]
20. Jaw, E.; Wang, X. Feature Selection and Ensemble-Based Intrusion Detection System: An Efficient and Comprehensive Approach. *Symmetry* **2021**, *13*, 1764. [CrossRef]
21. Gupta, N.; Jindal, V.; Bedi, P. CSE-IDS: Using cost-sensitive deep learning and ensemble algorithms to handle class imbalance in network-based intrusion detection systems. *Comput. Secur.* **2022**, *112*, 102499. [CrossRef]
22. Tama, B.A.; Comuzzi, M.; Rhee, K.H. TSE-IDS: A Two-Stage Classifier Ensemble for Intelligent Anomaly-Based Intrusion Detection System. *IEEE Access* **2019**, *7*, 94497–94507. [CrossRef]
23. Aldallal, A.; Alisa, F. Effective intrusion detection system to secure data in cloud using machine learning. *Symmetry* **2021**, *13*, 2306. [CrossRef]

24. Pelletier, Z.; Abualkibash, M. Evaluating the CIC IDS-2017 Dataset Using Machine Learning Methods and Creating Multiple Predictive Models in the Statistical Computing Language R. *Science* **2020**, *5*, 187–191.
25. Abbas, A.; Khan, M.A.; Latif, S.; Ajaz, M.; Shah, A.A.; Ahmad, J. A New Ensemble-Based Intrusion Detection System for Internet of Things. *Arab. J. Sci. Eng.* **2022**, *47*, 1805–1819. [CrossRef]
26. Pangsuban, P.; Nilsook, P.; Wannapiroon, P. A Real-time Risk Assessment for Information System with CICIDS2017 Dataset Using Machine Learning. *Int. J. Mach. Learn. Comput.* **2020**, *10*, 465–470. [CrossRef]
27. Gopalan, S.S.; Ravikumar, D.; Linekar, D.; Raza, A.; Hasib, M. Balancing Approaches towards ML for IDS: A Survey for the CSE-CIC IDS Dataset. In Proceedings of the ICCSPA 2020—4th International Conference on Communications, Signal Processing, and Their Applications, Sharjah, United Arab Emirates, 16–18 March 2021; Volume 2021-Janua.
28. Mhawi, D.N. Proposed Hybrid Correlation Feature Selection Forest Panalized Attribute Approach to advance IDSs. *Karbala Int. J. Mod. Sci.* **2021**, *7*, 15. [CrossRef]
29. Sekulić, A.; Kilibarda, M.; Heuvelink, G.B.M.; Nikolić, M.; Bajat, B. Random forest spatial interpolation. *Remote Sens.* **2020**, *12*, 1687. [CrossRef]
30. Feng, Q.; Liu, J.; Gong, J. UAV Remote sensing for urban vegetation mapping using random forest and texture analysis. *Remote Sens.* **2015**, *7*, 1074–1094. [CrossRef]
31. Alkasassbeh, M. An empirical evaluation for the intrusion detection features based on machine learning and feature selection methods. *J. Theor. Appl. Inf. Technol.* **2017**, *95*, 5962–5976.
32. Chen, S.; Webb, G.I.; Liu, L.; Ma, X. A novel selective naïve Bayes algorithm. *Knowl.-Based Syst.* **2020**, *192*, 105361. [CrossRef]
33. Huang, M.W.; Chen, C.W.; Lin, W.C.; Ke, S.W.; Tsai, C.F. SVM and SVM ensembles in breast cancer prediction. *PLoS ONE* **2017**, *12*, 161501. [CrossRef] [PubMed]
34. Schölkopf, B.; Platt, J.C.; Shawe-Taylor, J.; Smola, A.J.; Williamson, R.C. Estimating the support of a high-dimensional distribution. *Neural Comput.* **2001**, *13*, 1443–1471. [CrossRef] [PubMed]
35. Gou, J.; Qiu, W.; Yi, Z.; Shen, X.; Zhan, Y.; Ou, W. Locality constrained representation-based K-nearest neighbor classification. *Knowl.-Based Syst.* **2019**, *167*, 38–52. [CrossRef]
36. Thaseen, I.S.; Kumar, C.A.; Ahmad, A. Integrated Intrusion Detection Model Using Chi-Square Feature Selection and Ensemble of Classifiers. *Arab. J. Sci. Eng.* **2019**, *44*, 3357–3368. [CrossRef]
37. Ikram, S.T.; Cherukuri, A.K.; Poorva, B.; Ushasree, P.S.; Zhang, Y.; Liu, X.; Li, G. Anomaly Detection Using XGBoost Ensemble of Deep Neural Network Models. *Cybern. Inf. Technol.* **2021**, *21*, 175–188. [CrossRef]