Journal of Theoretical and Applied Information Technology
31st August 2022. Vol.100. No 16
© 2022 Little Lion Scientific

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

EFFICIENT CYBER INTRUSION DETECTION TECHNIQUE BASED ON AN ENSEMBLE CLASSIFIER

MD HARIS UDDIN SHARIF[1], SHAMIM UDDING AHMED[2]

[1]School of Computer & Information Sciences, University of the Cumberlands, KY, USA

[2]Dept: of Information Systems, Strayer University, MD, USA

E-mail: [1]haris.uddin.sharif@gmail.com, [2]shaamimahmed@gmail.com

ABSTRACT

In the digital world, online base network communication increased significantly. Cause of this, numerous cyberattacks occur every day and create a challenge for the network system to identify that intrusion on time. In addition, within the network, we cannot ignore the existence of intruders as they can launch many harmful cyberattacks. An intrusion detection system is the most valuable technique to help prevent network systems by investigating the network traffic. There are many intrusions detection research has been conducted. However, intrusion detection system still has some research gap and challenges that need to be improved to detect the accuracy of new intrusions. For this purpose, current artificial intelligence-based approaches, especially machine learning (ML) and deep learning (DL), are highly capable of intrusion detection in a network system. Both methods provide accurate outcomes. We proposed a novel, fast, efficient intrusion detection technique based on an ensemble classifier. The suggested classifier integrates all votes of the classifiers from the various instances that operate hard voting to reach the absolute voted class identity label. However, it has been shown on standard datasets of Network Security Laboratory and Knowledge Discovery in Databases (KDD). The suggested method acquired the most heightened precision of 98.19%.

Keywords: *Machine Learning; Deep Learning; Artificial Neural Network; Cyber Security; Intrusion Detection; Ensemble Method*

Proposed Acronyms

ADR	= Attack Detection Rate	IoT	= Internet of Things
ALDA-EC	= Aggregate Linear Discriminate Analyzed - Ensemble Classifier	KDD	= Knowledge Discovery Databases
		KNN	= K Nearest Neighbor
ANN	= Artificial Neural Network	KNNA	= K-Nearest Neighboring Algorithm
CFS-BA	= Correlation-based Feature Selection-Bat Algorithm	LASSO	= Least Absolute Shrinkage and Selection Operator
CNN	= Convolutional Neural Networks	LSTM	= Long Short-Term Memory
CVA	= Cross Validation Accuracy	ML	= Machine Learning
DDoS	= Distributed Denial of Service	MLA	= Machine Learning Algorithms
DL	= Deep Learning	MLP	= Multi-Layer Perceptron
DoS	= Denial of Service	NCF	= network connection features
DPA	= Deep Learning Algorithm	NFS	= Network Features Selection
DT	= Decision Tree	NN	= Neural Network
EC	= Ensemble Classifier	ODA	= Object Detection Algorithms
ECA	= Ensemble Classification Algorithme	PART	= Partial Decision List
FAR	= False Alarm Rate	PCA	= Principal Component Analysis
FN	= False Negative	Probe	= Probing Attack
FP	= False Positive	PSO	= Particle Swarm Optimization
FSS	= Feature Selection System	RFE	= Recursive Feature Elimination
GA	= Genetic Algorithm	R2L	= Remote to Local
HELAD	= A new incompatibility detection model called HELAD	RC	= Random Classifier
		RF	= Random Forest
IDS	= Intrusion Detection System	SNN	= Standard Neural Networks

SVM	= Support Vector Machine	TP	= True Positive
TML	= Traditional Machine Learning	U2R	= User to Root
TN	= True Negative	WEKA	= Waikato Environment for Knowledge Analysis
TNS	= Traditional neural networks		

1. INTRODUCTION

In the last decade, network security has become a critical study and the contemporary significance and progress in communication and internet technology development. These studies typically include IDS [1], firewalls, and antivirus software. Network security ensures safety from different types of cyber-attacks. These techniques play an essential role in defending the web from cyber-attacks. The high false alarm rate is the main problem of IDS, which in many cases poses a threat to the situation. The presented network problems can induce dangerous attacks to be neglected [2]. To decrease the ratio of False Alarms (FA) and grow the detection rate, researchers have developed various strategies that have been used in previous research. The 2nd issue with IDS is the weakness in identifying unexpected attacks that cause the network circumstances to change significantly. ML and DL-based approaches in intrusion detection can return outstanding performance. [3] focused that the Multi-Layer Perceptron (MLP) generalizes architecture's ability with related layers in attacking the Knowledge Discovery Databases (KDD) Cup 99. To classify the connection records of KDD Cup 99, [4] used support vector machines (SVM) and neural networks of classical machine learning classifiers. Jordan Architecture demonstrates an improved detection rate by introducing different experiments and cyber-attack rules [5]. The researcher suggested a joint model detect network intrusion based on tree-based algorithms [6]. There are utilized the NSL-KDD dataset to evaluate the system. For the cloud environment, there were proposed a network-based IDS [7]. They used the fuzzy c mean algorithm to identify intrusive network behaviors. Their clustering method's benefit was its ability to view the new attacks. The KDD99 dataset assesses a suggested method with an increased detection rate and lowers false positive alarms. [8] submitted an IDS using bagging ensemble techniques and boosting. There are use the tree algorithm as a base classifier. To evaluate their proposed method, they used the NSL-KDD dataset. The results showed that optimal performance was achieved using the Bagging Ensemble model and the J48 classifier when performing with a subset of 35 selected features related to false alarm rates and classification accuracy. The significant contribution of this method is given below:

1. We offer an ensemble-based classifier that authorizes other networks to work together to reach a common objective of intrusion detection classification on the NSL-KDD dataset.
2. Out of 41 parts, RFE selects 12 features.

This article introduces a novel strategy to resolve the problems of present network intrusion detection and protection technologies. It is established on Machine Learning Algorithms (MLA) but can be distinguished from different machine learning-based methods. Details in the following aspects:

1.1 Real-Time Attack Detection

To identify network attacks in real-time, many former experiments utilize non-machine-learning-based approaches. Therefore, billions of bits per second (Gbps) provide a much lower throughput than 100 Gbps. Although it uses a machine learning strategy, the proposed method may support real-time attack detection at traffic rates up to 100 Gbps. Machine learning in NIDS has become famous because it can detect unknown attacks. Cannot deploy machine learning-based algorithms on high throughput networks because they operate at prolonged speeds to handle Gbps traffic. To resolve this type of problem, experts suggest classifying two levels. One is for per-packet, and the other is for counter-flow detection [11].

1.2 High Attack Detection Accuracy

There are differences in the type and behavior of network attacks. And for this reason, it may not be adequate to use a single approach to identify different types of network attacks. Therefore, we can say that the suggested method can provide results using two-level attack detection. It can detect a few attacks by inspecting some packets. Still, in the case of many more attacks, the Distributed Denial of Service (DDoS) will be able to monitor the behavior of large network-wide streams. Hence, the method investigates individual packets and utilizes inter-flow, and in-inflow statistics flows to improve the precision of identifying attacks. The method could use two classifiers together, gaining much higher detection precision than ongoing work.

Therefore, due to the enormous improvement of computer networks and the quick rise of intrusion detection, Cyber security is becoming vulnerable daily. From academia and enterprise almost globally, the necessity of developing cyber security has attracted considerable attention. Even after utilizing various security applications such as user authentication, data encryption, malware prevention, and firewalls, businesses and many organizations have been the victims of recent cyber-attacks [12]. To hide within the method, attackers can exploit the vulnerabilities of the target system and launch different types of attacks. This way can leak critical information.

With the advancement of technology, the attacks threaten cyber systems' availability, integrity, and privacy. Therefore, intrusion detection systems (IDSs) [13],[14],[15],[16] need to be introduced. Introducing IDSs can protect the system from various attacks. IDSs are extensively placement on different disseminated systems to detect malicious intrusions. Which quickly responds to prevent infection and spread in the system. IDSs can generally be classified into two main categories - inconsistency and abuse detection [17]. Can make decisions based on the identification process. I designed anomaly detection to detect deviations from standard profile behavior to identify malicious actions. Similar IDSs perform better at detecting new types of attacks. Still, they couldn't avoid a high False Positive (FP) rate [18].

Furthermore, misuse identification can distinguish legitimate instances from malicious individuals [19]. Though it cannot detect unknown attacks, it does not even notice differences in acquaintances. This type of IDS is only reliable for detecting known attacks.

Unluckily, the attackers become more sophisticated, and new threats emerge simultaneously. Not only this, but vulnerabilities also appear rapidly. On the one hand, solving complex infrastructure problems increases the risk significantly in the short run. Besides this, the need for IDS to identify and deal with new attacks has been highlighted. Therefore, numerous methods have been studied and designed to enhance the detection rate and execution of IDSs. One of them is machine learning [20],[21],[22], which can apply to both anomaly and abuse detection models. IDSs are extensively placement on different disseminated systems to detect malicious intrusions. Which quickly responds to prevent further infection and spread in the system.

When a network is saturated with regular traffic flow, identifying an attack with a high Attack Detection Rate (ADR) while keeping a False Alarm Rate (FAR), only a fraction of the traffic may indicate malicious manners. One negative issue with the primary idea of requesting ML in IDS is that one classifier cannot be powerful enough to create a suitable IDS. Thus, investigators have grown up with the concept of creating ensemble classifiers for IDSs [23],[24]. Usually, the primary target of ensemble learning is to mix a separate classifier to make a more significant classification decision about the objects presented to the input [25]. For instance, an IDS dataset could create various classification performances to give training for a single classifier on different subsets. Therefore, an ensemble would average the result of multiple classifiers and thus become a more practical option.

In addition, the multiple attack categories and web traffic details pose other challenges for ML as they stretch the search position of the issue and management to high computational and time difficulty [26]. Notably, feature selection has been confirmed to be a satisfactory resolution for IDS, specifying highly appropriate features and eradicating useless ones with the slightest degradation of performance [27], [28]. There are three primary methods: wrapping, filtering, and embedded methods with feature selection. Ratio-based feature selection is one of the classical filter algorithms to gain information, where it represents a ratio of knowledge achieved to the report. It reduces bias towards many valuable features and solves data retrieval errors. However, the summit may be biased towards parts with elements used in some cases are noticed four components. Unlike the data gain ratio, the selection input properties make the most of the output and the correlation-based properties. It also helps to reduce the redundancy of selected features. This algorithm prefers one feature time condition to confirm robust relation with results, which can flexibly execute in both parts.

This article proposes an effective practical, precision intrusion detection system to detect different attacks. First, a nature-inspired feature selection algorithm is presented as a traditional means of dimensionality decrease and redundancy elimination to retrieve a subset of the original features. Second, this system's inconsistency between normal and malicious traffic harms attack detection accuracy. To overcome this problem, the solution uses Ensemble Classifier to reduce the bias between different training datasets. Feature selection and ensemble classifiers are mixed to enhance the strength and precision of IDS with less calculated time. Finally, a neutral model can be created that can detect both famous and rare intruders. The main contributions of the work are presented below:

- We are proposing a new approach that works. This method merges the benefits of feature selection and ensemble classifier to achieve appropriate intrusion detection.
- CFS-BA-based methods are provided for feature selection. This method is used to evaluate the correlation of selected features. This method is also effective for optimizing the training and evaluation stage.
- An ensemble method is present to enhance the multi-class classification execution on unstable datasets by integrating determinations from multiple classifiers into one by using a voting classifier founded on the Average of Probabilities (AOP) combined rule.

2. LITERATURE REVIEW

IDS continually attracts the research society's concentration as an effective tool in computer-based techniques for confirming cyber security. Many new solution methods have been proposed to enhance the performance of IDS. Only Machine Learning-based IDS functions are considered here, which use the ensemble classifier, mainly focusing on the hybrid approach [29].

ML-based Network Intrusion Detection Systems (NIDS) have been continuously improved. There were limitations to accurately identifying different network attacks with a single machine learning algorithm. So, NIDS research using various machine learning algorithms is actively underway.

The ML-based IDS is split into a packet-based approach that utilizes packet data. A session-based system uses session data to teach the method based on the class of data utilized packet-based method to convert primary packet data into machine learning features. The convolutional neural networks (CNN) perform learning and classification using conventional machine learning algorithms. It is not necessary to determine the characteristics of the complex neural network CNN before training. Therefore, it simplifies the training process without any intervention.

When processing a new session, the session-based method generates datasets to create features of the session properties inside the IDS. Then processes session packets and update session data. After processing the final package, it develops components for the session by processing the last corrected data, and then these features are utilized in machine learning. This allows for a small number of statistical values without using many packets in one session. The number of training and classification is minimal. Since it is very effective for quick training and classification, it can handle significant network traffic.

According to the number of algorithms used, single and multiple classification-based NIDSs can be classified into four types. Here each method is tested.

Table 1: Classification of ML-based NIDS according to classifiernumber and feature type.

	Single classifier	Multiple classifiers
Packet-based	[30][31]	[32]
Session-based	[33][35, 36][37-39][40-45]	[45][47, 48]

2.1 Packet-Based Single-Machine Earning Algorithm Method

These systems learn and categorize packet data through a single MLA. It has an advanced feature to detect malicious code in packet payload data. Under the conventional signature-based NIDS method [30], Table 1 shows individual packets are analyzed independently to determine if an attack has occurred. However, this approach benefits from detecting an attack in real-time. Besides, there is a risk of a zero-day attack, alternate attack, and bypass in this case. Recently, an attack detection method has been proposed to overcome these vulnerabilities by collecting multiple packets instead of a single package.

2.2 Packet-Based Multiple-Machine Learning Algorithm Method

This approach identifies attacks utilizing numerous MLA instead of a single packet-based data algorithm. So, it can execute training and classification more efficiently than a single algorithm. However, MLA should generate many features from packets, such as the packet-based single machine learning algorithm method. Therefore, it isn't easy to use on large networks due to prolonged training and classification speeds [33].

2.3 Session-Based Single-Machine Learning Algorithm Method

This method extracts the features for a session instead of using packets. It is commonly applied to single algorithms for training and classification [40 - 45]. This method is the most common study in elementary machine learning-based writings. This method does not use packet data and produces several packet sizes or numbers in a session. It uses very little memory. In particular, it uses a single algorithm and processes a small number of features, which can speed up training and classification. So, it applies to large networks (including heavy traffic). However, it is challenging to provide high detection rates for different attacks using a single algorithm. Further, features are usually created after the end of the session, so possible attacks are completed before it is detected. This is the most intense limitation of this category.

2.4 Session-Based Multiple-Machine Learning Algorithm Method

This method uses different classification algorithms to perform training and classification using features for a session simultaneously. This category has several types: the ensemble and the multi-layered system [46],[47]. The ensemble method consolidates the results and can apply several algorithms as well. It uses different algorithms for different classes to improve detection performance. After implementing a specific algorithm, one must wait for the multi-level approach's results.

In most cases, it turns out that it uses both supervised learning and supervised learning simultaneously. It uses the Decision Tree (DT) to create partitions and then uses the k-nearest neighbor (KNN) to generate each division. The session-based numerous machine learning algorithm approach has a very high classification execution. However, it is challenging to support real-time attack detection due to multiple machine learning algorithms. In reality, it is impossible to process network traffic in real-time.

Furthermore, since the overall implementation cost of this method is high, it is pretty challenging to implement network security measures in this method. Different approaches have been adopted to improve detection accuracy and speed. However, there is minimal research on intrusion prevention systems to detect and protect against real-time attacks. Thus, research on real-time intrusion prevention systems is needed [11].

The feature selection technique [58],[59] for reducing computational complexity; is used as a pre-processing step in ML algorithms. The main goal of IDS is to eliminate its functionality and even irrelevant features. Hota and Shrivas [27] proposed a model to obtain a more robust and effective classifier that utilized various feature selection techniques to remove exterior features. The results displayed can achieve maximum accuracy for the C4.5 NSL-KDD dataset with data gain. In this case, only 17 elements have been considered. In addition, Khammasiyand Christian [28] has applied Network IDS as a search algorithm and logistics regression as a learning algorithm to select the best subset. The results show that their method provides a high detection rate with only 18 features for KDDCup'99 and 20 features for the UNSW-NB15 dataset. Abdullah et al. [48] also presented a framework of IDS with elements within the NSL-KDD data set based on splitting the information dataset into various sub-sets and merging them utilizing the Information Gain (IG) filter.2.2. on ensemble classification.

However, ensemble models are machine learning approaches that mix several primary models to decrease false-positive rates and develop more accurate solutions than a single model. Gaikwad and Thool [49] presented a bagging ensemble method utilizing REP Tree as its primary classifier, bringing less time to create the model and delivering the highest classification accuracy with the lower false positives on the NSL-KDD dataset. Jabbar et al. [50] have suggested a cluster-based ensemble classifier for IDS consisting of an Alternating Decision Tree (AD Tree) and K-Nearest Neighboring Algorithm (KNN). The practical outcomes display that the presented ensemble classifier outperforms other existing precision and detection rate strategies. Paulauskas and Auskalnis [51] proposed an ensemble model of four different base classifiers called J48, C5.0, Naive Bayes, and Partial Decision List (PART), depending on the concept of multiple integrations. Through this, it is possible to create a strong student. It is proved that their ensemble model creates more authentic outcomes for an IDS. Moustafa et al. [52] proposed new statistical flow features and developed an AdaBoost ensemble learning method to detect attacks effectively.2.3. to mitigate adverse events, particularly botnet attacks in the Internet of Things (IoT) networks. To improve the execution of IDSs, multiple hybrid methods utilizing both feature selection and ensemble methods have been developed. Malik et al. suggested a hybrid approach of Particle Swarm [53] Optimization (PSO) and Random Forest (RF). More relevant features for each class support the proposed model and produce a higher accuracy and low false-positive rate than other algorithms. Pham et al. [25] produced a combination model that uses the gain ratio approach as feature selection and bagging to mix tree-based base classifiers. Practical outcomes showed that the bagging model produced the best performance, which worked on a 35-feature subset of the NSL-KDD dataset and used J48 as the base classifier. Also, Abdullah et al. [48] have developed an IDS that uses IG-based feature selection and ensemble learning algorithms. Tests on the NSL-KDD dataset show maximum accuracy when utilizing RF and PART as base classifiers under

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

product probability rules. Moreover, Salo et al. [27] demonstrated a hybrid IDS which combines the feature selection techniques of PCA and IG with an ensemble classifier formed by SVM, IBK, and MLP. The IG-PCA-Ensemble method performs better than most existing methods, as evidenced by a comparative analysis of several IDS datasets. Khan et al. [65] proposed a scalable and hybrid IDS that outlines the causes of large-scale data generated from the extensive network infrastructure. Which separately employs incompatibility and abuse detection based on the Spark ML and Convolutional-LSTM (Conv-LSTM) networks. Also, based on the damped incremental status-tax algorithm for feature selection, Zhang et al. [54] proposed a new incompatibility detection model called HELAD. It is a biological integration of multiple deep learning strategies for classification [10]. In [55], a novel IDS established on diverse feature selection and two-level classifier ensembles has been suggested. Experimental outcomes show significant developments in the detection rate on the NSL-KDD and UNSW-NB15 datasets [29].

A successful intrusion consists of the following steps [56]:

- Reconnaissance: Act to analyze significantly to acquire data about the target. It is performed by executing network commands such as “nslookup, and” “whois” to get a domain name, IP addresses, server details, etc.
- Scanning and probing: Finding unsure areas in the target method to search for valuable data.
- Attack from Remote to Local (R2L): R2L means gaining access by sending malicious network packets from a distance so that the attacker can use commands on the target system. They are performed using open ports, password prediction, sniffing, etc., taking advantage of system vulnerabilities.
- Attack of User to Root (U2R): These attacks are performed to gain access to the system administrator. As a result, an attacker can take complete control of the system.
- Start Attacks: Attacks are induced after U2R is successful. The best example of this is stealing or altering confidential information.

IDSs are assigned to raise the alarm if such activity happens in internal networks. Effective IDS are generally developed using machine learning and data mining techniques because these can significantly determine intrusions. These methods have a training stage in which the model is trained utilizing datasets. Datasets include tagged samples of both attack and standard class. After teaching these models with mathematical algorithms, the oriented model experiments on different examples of data to check the precision of projection [57].

3. RESEARCH GAP

NSL-KDD dataset is an enhanced form of the KDD cup99 dataset [48]. NSL-KDD is proportional to 10% of the KDDcup99 dataset because it consists of vast information. The examples in the dataset are classified as an attack or standard. Different machine learning models have been created and studied utilizing the NSL-KDD dataset and discussed in [58]. To enhance the data management abilities of IDS and deal with the enormous size of the NSL-KDD dataset, multicore hardware platforms appear to be appealing. Hardware acceleration CNN is used for image processing applications [59]. In this case, all changes are performed on the KDD cup dataset:

- Unnecessary records have been removed, so the classifier does not show neutral results.
- There are multiple records in the training and testing datasets whose performance is reasonable.

Each sample of the NSL-KDD dataset is composed of 41 features. Classification features are usually called an attack or normal. Names, data types, descriptions, etc. [58] are called attribute descriptions. The NSL-KDD dataset contains 39 kinds of attacks. All classes are divided into DOS, Probe, U2R, and R2L.

Observations show that 39 different types of attacks took place here. There are ten types of attacks in the Denial of Service (DoS) class, 6 in probe class, 16 in R2L class, and 7 in U2R.

4. METHODOLOGY AND EXPERIMENTS

4.1 Feature Reduction Methods

Most data mining approaches and machine learning could not perform correctly with intrusion detection cause of the dataset’s huge complications and size. These processes take enormous time to classify attacks, creating performance difficulties in real-time environments. It causes many features of in-network data to be prepared by the Intrusion Detection System. Quantity and quality of features matter for more helpful classification, and it helps us recognize their urgency and co-relation. The classification quality will be reduced if the selected features are too few, while on the other hand, it will damage the generalization if they are more than required. In Intrusion Detection Systems, experimental outcomes show that feature extraction techniques enhance calculated cost and precision [12]. Improve accuracy and decrease the time for attack detection, dimensionality, and feature reduction strategies are used as a pre-processing step.

4.2 Feature selection

The feature selection process is used automatically or manually to select the features (i.e., individual variables). Those are very much important in providing desired forecast results. One of the fundamental ideas of machine learning is feature selection, which significantly influences the model's performance. Maintaining unessential features in the dataset can reduce models' precision and make the model learn based on non-relevant features. Figure 1 shows the feature selection process in detail.

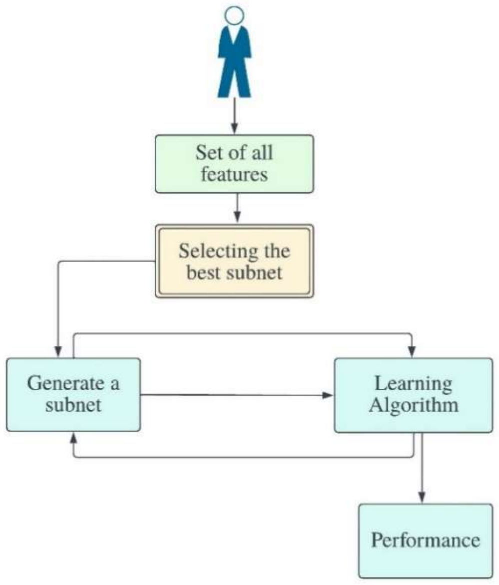

Figure 1: Feature Selection Method

4.2.1 Benefits of Feature Selection:

Minor data duplication means less reliance on noise when making decisions.

Minor deceptive data indicates modeling accuracy and improvement.

To find a shorter set of attributes, the feature selection methods are used to enhance the general result of the process and generate minor mistakes. Another objective is to reduce calculation time and storage uses. To improve the precision of attack detection, feature selection techniques are used. Elhag et al. [13] described that the Principal Component Analysis (PCA), Information Gain (IG), and Genetic Algorithm (GA) are the dominant feature selection methods. There are two feature selection approaches, i.e., Filter and Wrapper, including various FS methods.

Filter methods are relatively robust against overfitting. It does not use any classifier to assess features. It utilizes autonomous guessing techniques, for example, measurement of distance, consistency, and interrelationship.

4.3 Filter Method:

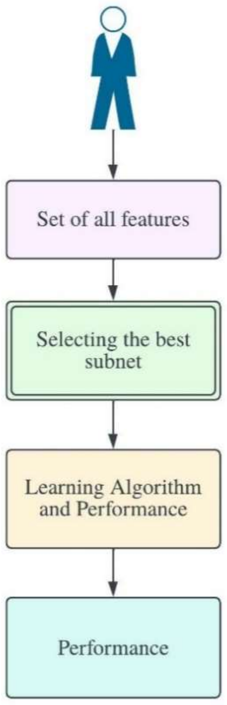

Figure 2: Filter Method

This method filters by name and accepts only subsets of relevant features. After selecting features, this model is created. Utilizing a correlation matrix, the refining process is completed, and using Pearson correlation, most typically finished. Figure 2 shows details.

First, the heat map of the Pearson correlation is plotted, and the correlation of the independent variable with the output variable called MEDV is imagined.

Only special features have a correlation of 0.5 above the output variable. Here the value of the correlation coefficient ranges from -1 to 1. The results obtained can be related as follows:

- Weak correlation: Value close to 0 (correct 0 means no relation).
- Strong positive relationship: Value close to 1
- Strong negative relationship: Value close to -1

The classifier is used as a black box in the wrapper technique to evaluate the best features. These methods perform significant hypotheses yet go through high dimensionality due to the calculated cost of developing the classifier from time to time.

4.4 Wrapper Method:

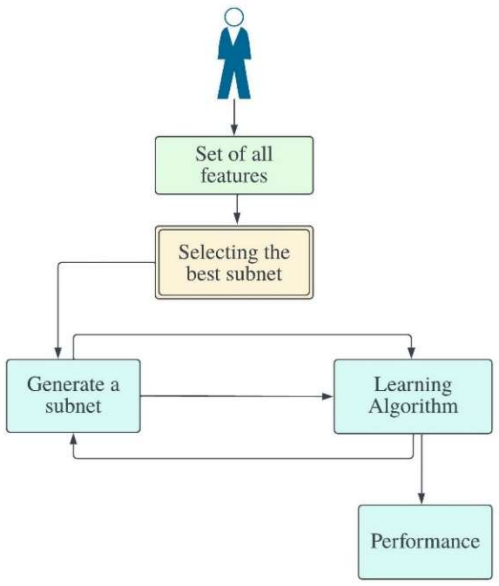

Figure 3: Wrapper Method

Machine learning algorithm needed for wrapper method. The wrapper method is used to evaluate the algorithm's performance, i.e., feed the features to the selected Machine Learning algorithm and add or remove the features based on the model performance. Although it is a repetitive and computationally expensive process, it is more efficient than the filter method. Figure 3 describes the wrapper method accordingly.

4.5 Forward Selection

This iterative method initially must begin with selecting a single feature. Overall performance is monitored before adding the next part of the essential features need to be added until the best results are found. Attempts are made to increase the performance of the model by adding features.

4.6 Backward Elimination

The training model adds all the features to start and will remove one part in the next iteration. It is precisely the reverse to deliver elimination. It will remove the least significant feature in each iteration to improve precision. Features need to be removed one by one while improvements continue.

4.7 Embedded Method

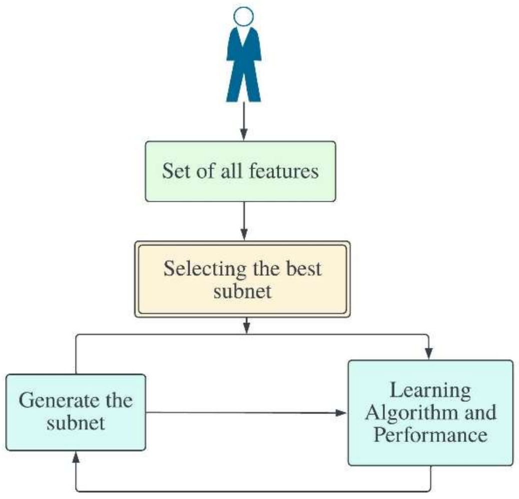

Figure 4: Embedded Method

In each iteration, embedded methods carefully extract those features which donate numerous to projection while training the technique. Standardize the methods generally utilize embedded methods that penalize a feature with a measure threshold.

Figure 4 shows that embedded techniques are a mixture of filter and wrapper methods. The method is executed by algorithms that have their required feature selection methods. Few of the well-known embedded methods like LASSO and RIDGE regression are utilized to decrease the problem of overfitting by penalization.

Lasso regression L1 achieves regularization where fines equivalent to the absolute value of the coefficients are added. Ridge regression performs L2 regularization where penalties equal to the square of the dimensions of the coefficients are added.

Finding a subset of attributes from the set is the primary goal of the feature selection, which adequately conveys the data, and the features of the subset are appropriate to the projection. Feature selection methods can be mainly classified as wrapping, filtering, and embedded approach [60]. While filter methods evaluate the relevancy of the features from the dataset and the selection of the features is founded on the statistics, the classification execution is utilized in wrapper approaches as a part of the featured subsets assessment and selection procedures. In contradiction to wrapper approaches, embedded systems are less intensive cause they include relations between feature selection and learning. Embedded methods integrate a regularized risk function to optimize feature designating and predictor parameters [61]. It is not easy to modify the classification model [62].

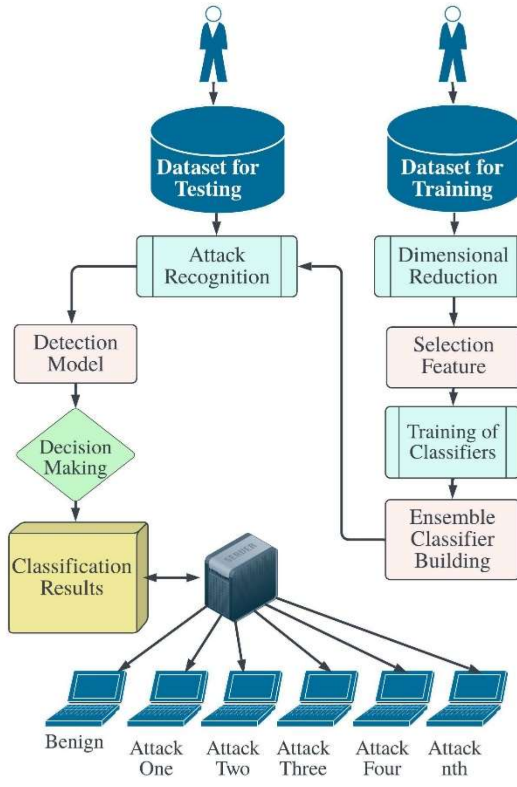

Figure 5: The framework of the proposed Feature Selection-Ensemble model

4.8 Feature Extraction

The rows in a dataset represent the sample, and the columns represent the features where the features are displayed in quantitative and thematic search results. To decrease the dataset's dimensionality, feature extraction is utilized by reducing the set of attributes so that the precision of attack identification is not altered and the time used in discovery is diminished. Many feature extraction methods can be observed, such as - self-organizing maps, fundamental component analysis, etc.

The ensemble-based classifier model (Figure 5) utilizes majority votes for class decisions to classify samples into Intrusion Detection systems (IDS) contaminated or non-contaminated classes. The most positive sample of base classifiers was classified as non-contaminated IDS classes. Similarly, most of the negative sample votes in the base classifiers are classified as malicious IDS classes. The results of all the base classifiers of the Ensemble method are combined into a significant classification to increase the effectiveness of the classification. The Ensemble-based Classifier model is shown in Figure 6 [63].

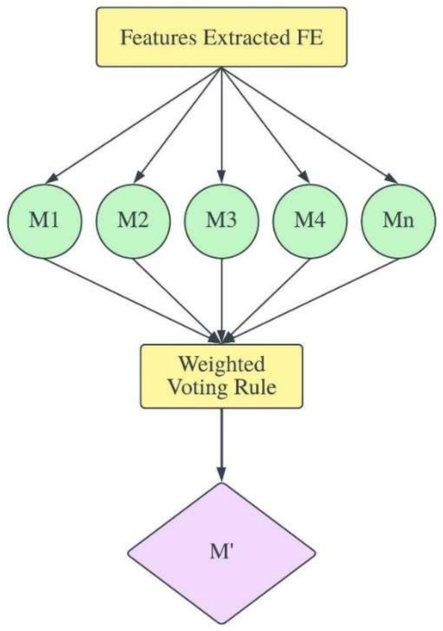

Figure 6: Block diagram of the Ensemble-based Classifier model

Let us think of a training set of size "with" achievable random examples; then, the ensemble methods mix a series of "learned models or random examples" to develop an enhanced model. Delivered a training set "(i.e., extracted features" utilizing ALDA-EC) utilizing "tuples, each iteration", a training set "or" of "tuples is sampled with replacement from". A classifier model "is learned from each training set" [63].

Table 2: Algorithm 2 Ensemble-based Classification

```
Input: Features Extracted 'FE' samples
Output: Ensemble classification
1: Begin
2: For each extracted feature, 'FE' and '$k = 1$.'
3: For each class '$c$' and samples'
4: Evaluate the Weighted Voting Scheme using (7)
5: Measure KNN distance classification using (9)
6:     Measure total vote using (10)
7:    Assign the highest vote
8:            End of
9:        End of
10: End of
```

Ensemble-based classification begins with the extraction features obtained using ALDA-EC given in pseudo-code. This is followed for each class and sample. Each repetition includes two steps. The first step is to apply the weighted voting scheme to reach the complex instance classification. Each instance is classified by calculating the total votes founded on the K Nearest Neighbors (details in

Table 2). Then, where the final classification represents a simple majority vote, the maximum number of votes cast for each sample is considered. Thus, improving the accuracy of IDS classification relies on robust encrypted classifiers, bootstrap aggregation, and k-nearest neighbors.

4.9 System Overview

The binary classification NSL-KDD dataset is the goal of the proposed system. Figure 7 illustrates the overall pattern shown in the flow diagram. First, Pre-processing and data cleaning are used for the data group. Then, the feature is selected after that dataset is passed through various machine learning classifiers for classification; finally, the ensemble classifier is utilized to reach higher precision. In the next section, each step of the proposed system will be described in detail.

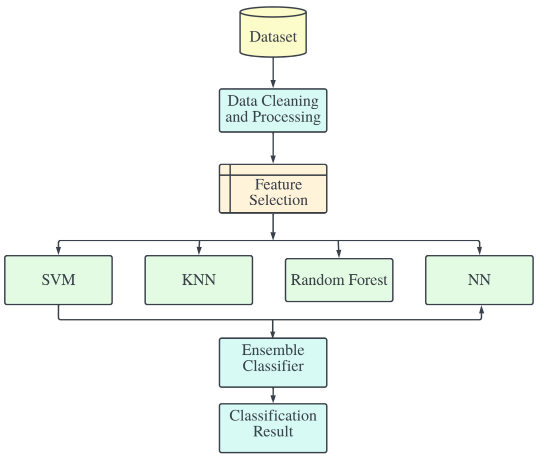

Figure 7: Proposed system workflow diagram.

4.10 NSL-KDD Dataset

The NSL-KDD [37] was suggested to overcome the weaknesses of the KDD99 dataset. Based on the KDD99, carefully selected the NSL-KDD records. To avoid the classification bias problem, descriptions of different classes are balanced in the NSL-KDD. The NSL-KDD also deleted duplicate and excessive records, including only an average number of records. Therefore, can implement the experiments on the entire dataset. The NSL-KDD relieves the issues of data bias and surplus data redundancy to some degree. So, the NSL-KDD doesn't enclose new data; thus, small class examples are still indifferent, and its models are always out-of-date.

The NSL-KDD dataset [9] was suggested in 2009 as a new corrected version of the actual dataset KDDCup'99 [64]. On the one hand, NSL-KDD keeps the challenging and beneficial features of KDDCup'99. On the other hand, it has solved some of the errors inherited from the original dataset by justifying the number of examples, eradicating excessive records, and managing the variety of the selected model. It is noteworthy that complying with the NSL-KDD datasets maximizes the disadvantages of projection, which establish its excellent features. The primary dataset was evaluated using several benchmark classifications to group the records into five difficulty levels—annotating each example with the number of its fruitful projections [66]. For each problematic level group, the quantity of specified data is inversely balanced with the record percentages from the original KDDCup'99 dataset.

Each instance of the NSL-KDD data set included 41 features. The classification feature names the attack or normal. Name, data type, and description are the three characteristics described [67]. There are 39 attack types in the NSL-KDD dataset, categorized into four attack classes, i.e., DOS, Probe, U2R, and R2L.

Observations show that 39 different types of attacks took place here. Ten types of attacks in the Denial of Service (DoS) class, 6 in probe class, 16 in R2L class, and 7 in U2R. A summary of data examples of the NSL-KDD dataset used in the study and training stage of the generated IDS model is shown in Table 3 [57].

Table 3: NSL-KDD Dataset Overview

DATA SET TYPE	Number of Instances					
	Total Instances	Normal Instances	DoS Instances	Probe Instances	U2R Instances	R2L Instances
Training set	125973 %	67343 53.46	45927 36.45	11656 9.25	52 0.04	995 0.79
Test set	22543 %	9711 43.08	7458 33.08	2421 10.74	200 0.89	2754 12.22

Table 4: Features of the NSL-KDD dataset

#	Feature	#	Feature
1	duration	22	is guest login
2	protocol type	23	Count
3	service	24	srv count
4	flag	25	serror rate
5	src bytes	26	srv serror rate
6	dst bytes	27	rerror rate
7	land	28	srv rerror rate
8	wrong fragment	29	same srv rate
9	urgent	30	diff srv rate
10	hot	31	srv diff host rate
11	num failed logins	32	dst host count
12	logged in	33	dst host srv count

13	num compromised	34	dst host the same srv rate
14	root shell	35	dst host diff srv rate
15	su attempted	36	dst host same src port rate
16	num root	37	dst host srv diff host rate
17	num file creations	38	dst host serror rate
18	num shells	39	dst host srv serror rate
19	num access files	40	dst host rerror rate
20	num outbound cmds	41	dst host srv rerror rate
21	is host login		

Table 5: Attacks Categories with its types of the NSL-KDD dataset

Attack Category	Attack Name
DoS	Back, Neptune, Land, Pod, Smurf, Udp-storm, Teardrop, Apache2, Worm, Pro- constable.
Probe	Nmap, Ipsweep, Portsweep, Saint, Mscan, Satan
R2L	Ftpwrite, Warezmaster, Httptunnel, Guess- Password, Phf, Warezclient, Snmpguess, Multihop, Xlock, Spy, Xsnoop, Sendmail, Snmpgetattack, Imap, Named
U2R	Rootkit, Loadmodule, Xterm, Sqlattack, Perl, PS, Buffer overflow

There are 41 attributes in the NSL-KDD dataset (Table 4) for each connection record, with class labels having attack types. The class of attacks is classified into four attack classes (Table 5 and Figure 7) [9].

1. **Denial of Service (DoS):** Denial of service (DoS) is one kind of cyber-attack. Where the attacker directs the flow of traffic requests to a system, this allows the computing or memory resource to be too busy or too full to handle legitimate requests. In this method, an honest utilizer is denied access to a machine.
2. **Probing Attack (Probe):** A probing attack is a new threat to the intrusion detection system. The probe attack is deliberately created so that the attacker can identify the target and generate a report with a recognized "finger drop."
3. **User to Root Attack (U2R):** In this attack, the adversary can access a standard user account on the targeted method (obtained by password inhaling, utilizing a dictionary, or social engineering attack) and exploit some system vulnerabilities to gain access to the route.

Remote to Local Attack (R2L): An attacker can send packets to the targeted machine through the network. An attacker exploits some vulnerabilities to gain local access as a device used on which the attacker has no account.

4.11 Support Vector Machine (SVM)

SVM is one of the standard MLA, like a biased classifier described by a remote hyperplane. That means algorithms design the best hyperplanes for labeled training data to classify new inputs. A two-dimensional hyperplane is a line that divides space into two parts. Support Vectors are the coordinates of separate inspections.

Several possible hyperplanes could be chosen to separate the two classes of data points. Must select the hyperplane, including a maximum margin. Here top margin means the entire length between the nearest data points of both classes. There are some criteria to specify the proper hyperplane [68].

4.11.1 Criterion 1

Three hyperplanes have been tried to separate the two classes, Figure 8 [68].

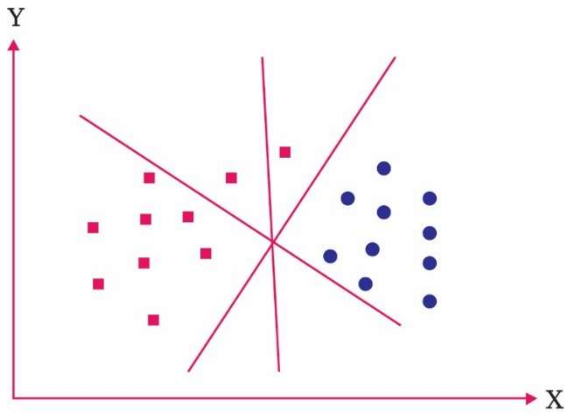

Figure 8: Criterion 1

To choose a hyperplane that can separate the two classes. It turns out that Hyperplane X meets this standard [68].

4.11.2 Criterion 2

All hyperplanes are dividing two classes; now, the question is how to identify the correct one [68]?

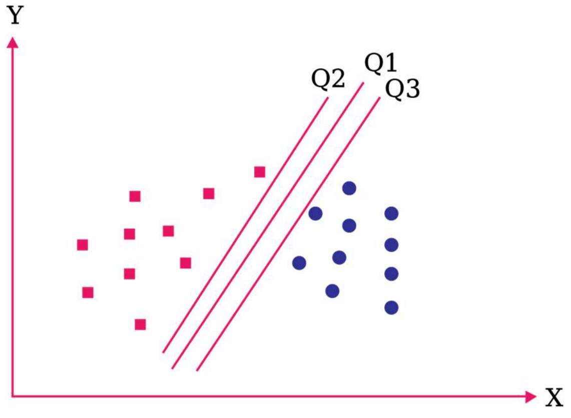

Figure 9: Criterion 2

Here one must consider the nearest data point of both classes and the maximum distance between the hyperplanes. This distance is called the 'margin.'. The image above shows a top length of Q1 from both classes' nearest points (Figure 9 [68]).

4.11.3 Criterion 3

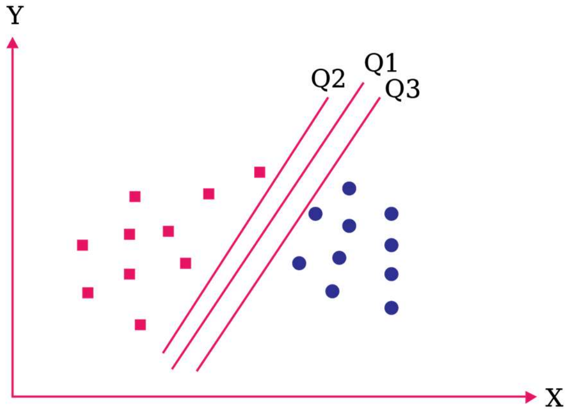

Figure 10: Criterion 3

In this measure, if we prefer hyperplane Q2 according to a higher margin than Q1, it misclassified the data points. So, hyperplane P2 has classification mistakes, but hyperplane Q1 can classify accurately Figure 10 [68].

4.11.4 Criterion 4:

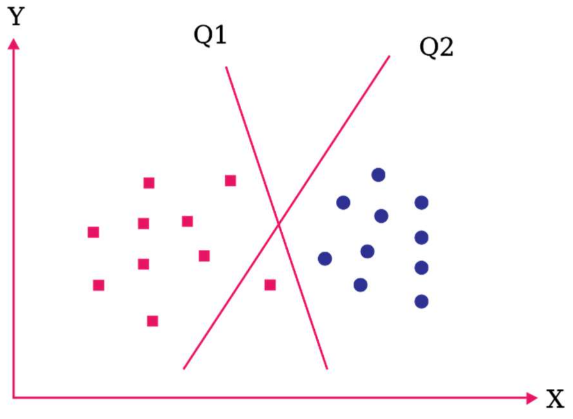

Figure 11: Criterion 4

SVM has the property to supervise the outliers. It is a robust algorithm in the case of outliers Figure 11 [68].

4.11.4 Criterion 5:

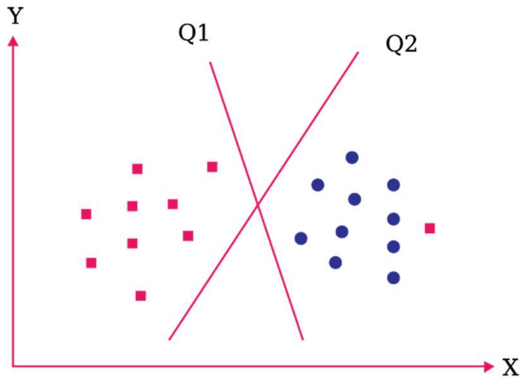

Figure 12: Criterion 5

Now, handling this criterion is challenging in using a single line as a hyperplane. SVM operates this problem by utilizing more features. It can use third plane Z, besides X and Y planes, including Figure 12.

$$z = x^2+y^2$$

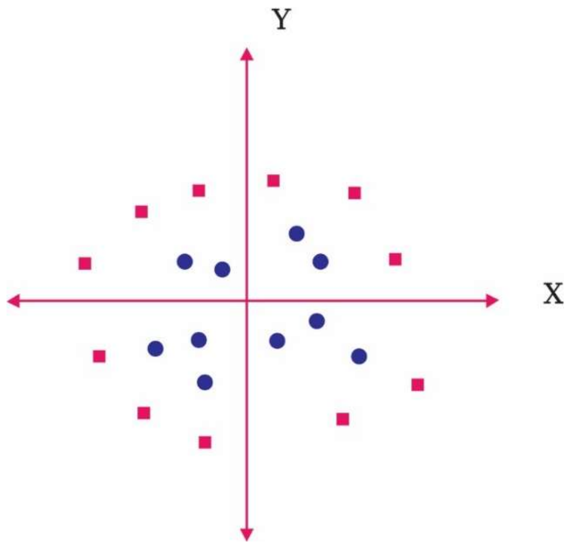

Figure 13: Criterion 5

As a plan for the data points across X-Z planes, we get the above diagram that displays the two classes' segregation. SVM can manage the split of various categories of data points with suitable hyperplanes. In the SVM model, some parameters are needed to be defined to turn for the systematic working of that model Figure 13 [68].

A relatively simple supervised machine learning algorithm is a support vector machine (SVM) used for regression and classification. Sometimes SVM is beneficial for degenerates. However, it is preferred for classification because it finds a hyper-plane that creates a boundary between the data types. This hyper-plane is nothing more than a line in 2-dimensional space. In SVM, each data item is plotted in a data set in an N-dimensional area. The data's number of features/attributes is N. To separate the data find the optimal hyper-plane. So, SVM can choose between two classes, i.e., it can only perform binary classification. However, there are different strategies to use for multi-class problems.

4.12 D. K Nearest Neighbor (KNN)

K Nearest Neighbor (KNN) is a led machine learning practical algorithm for classification problems. It is a lazy and non-parametric learning algorithm which means no guesswork for the distribution of primary data. Non-parametric means there is no calculation for the underlying data distribution. In KNN, K is the number of nearest neighbors. In KNN, K is the digit of closest neighbors. The number of neighbors is the primary deciding factor. It estimates the length between the test data and the input and gives the forecast accordingly. KNN model by following the below steps. We can execute a KNN model by following steps [69]:

Step-1: Loading the information
Step-2: Initialize the value of k
Step-3: For obtaining the expected class, repeat from 1 to the total number of training data points

a. Estimate the length between test data and each row of training data. Utilize Euclidean distance as a distance metric since it's the famous method. The other metrics that can use are Chebyshev, cosine, etc.
b. Sort the estimated distances in ascending order established on distance values
c. Get the top k rows from the sorted array
d. Get the most recurrent class of these rows
b. Replace the expected class.

4.13 Random Forest

The Supervised classification algorithm is called the Random Forest algorithm. We can notice it from its name, that is, to make a forest in some way and create it randomly. There is a straight connection between the number of trees in the forest and its outputs, i.e., the bigger the number of trees, the more exact the outcome. But one thing to state is that making the forest is not precise as creating the judgment with the data attain or attain index method.

4.14 Classification in random forests

To attain the results, classification in random forests utilizes an ensemble methodology. To train various decision trees, the training data is fed. Will randomly select observations and properties in the dataset during node partitioning.

A rainforest method depends on different decision trees. The decision tree comprises the decision node, leaf node, and root node. The final output produced by a particular decision tree is the leaf node of that tree. The majority-voting system followed a selection of the outcomes. In this case, the output selected by most decision trees is considered the final output of the rainforest system. Figure 14 shows an easy random forest classifier [70].

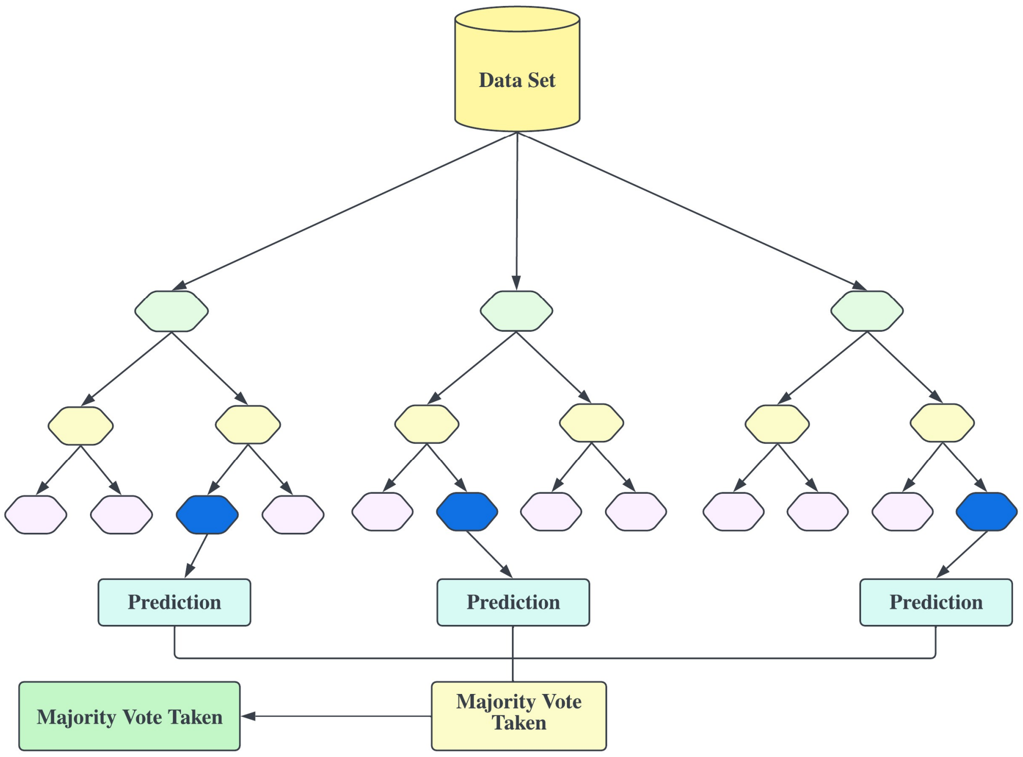

Figure 14: Classification in random forests

4.15 Random Forest Classifier

Another decision tree technique was proposed by Bremenis [35] as a random forest. It works by constructing multiple decision trees. It takes thousands of input variables without deleting and classifying them based on their importance. RF can be explained as an ensemble of classification trees where each tree donates a single vote for the work of the most numerous classes to the input data. When running RF, fewer parameters need to be specified than other machine learning methods, such as support vector machines, artificial neural networks, etc. An RF can define as a group of individual tree-structured classifiers:

$$d(g, \delta_m), \qquad m = 1 \dots i \dots \}$$

Here d focuses on Random Forest classifier, $\{\delta_m\}$ for random vectors, g = input variable. Mainly, Random Forest has a small computational gravity, and it is thoughtless to the elements and outliers. Further, over-fitting is a tiny part linked to a specific decision tree, and no requirement to crop the trees, t; that is a vast job [71].

4.16 Artificial Neural Network

A multi-layer fully connected neural net is an Artificial Neural network (ANN) that looks like the figure below. There are three layers, i.e., input layer, multiple hidden layers, and output layer. Every node is in one layer, and each node is connected to every node in the next layer. We make the network deeper by increasing the hidden layers (Figures 15 & 16).

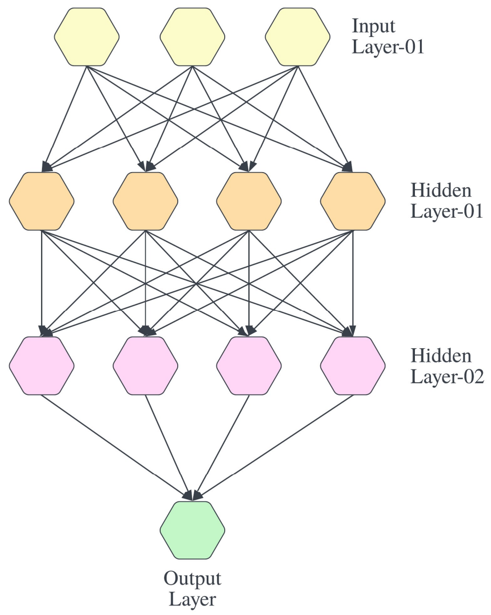

Figure 15: Artificial Neural Network

Will encounter this if you zoom in on one of the hidden or outcome nodes shown in the figure below.

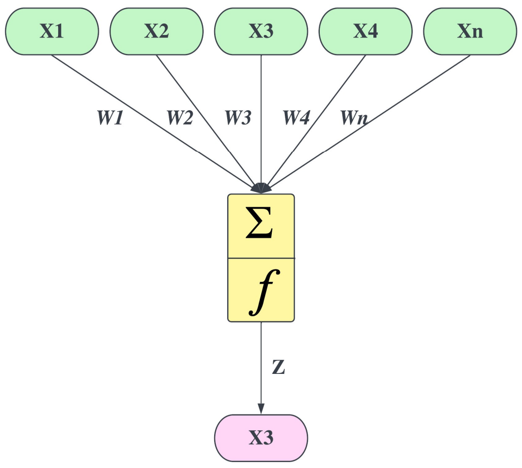

Figure 16: Artificial Neural Network

A provided node handles its inputs' weighted sum and gives it through a non-linear function which activates the system. This is the outcome of the node that then evolves the information of other nodes in the subsequent layer. The result is estimated by performing this process for all the nodes. The signal moves from left to right. Introducing this deep neural network represents knowing the importance connected with all the end.

The equation for a provided node looks as follows. The calculated total of its information gave through a non-linear activation process. It can be described as a vector dot output, where n is the digit of data for the node.

$$Script\,p = f(u.v) = f\sum_{i=1}^{n}(u_i v_i)$$
$$u \in q_{1\times n},\ v \in q_{n\times 1},\ p \in q_{1\times 1},$$

Due to simplicity, I skipped the bias term. Bias means giving input to all the nodes and always presenting the value 1. It authorizes moving the output of activate function to the right or left. It also favors the teaching approach when all the input features are displays difficulty right now, safely skip the bias conditions. The above equation follows for completing the bias formed [72].

$$p = f(a + u.v) = f\left(a + \sum_{i=1}^{n}(u_i v_i)\right) u$$
$$\in q_{1\times n},\ v \in q_{n\times 1},\ a \in q_{1\times 1},\ p$$
$$\in q_{1\times 1},$$

So far, told the forward pass, meaning provided an input and values how the result is calculated. After the teaching is complete, there are need only to run the forward pass to create the projections. But need to teach the model to learn the importance, and the training method works as follows [72]:

- Arbitrarily starts the significance for all the nodes.
- There are intelligent initialization processes that will research in another paper
- For each training instance, execute a forward pass utilizing the existing importance and estimate that each node going from left to right. The result is the worth of the last node.
- Compare the result with the exact target in the training data and estimate the error utilizing a loss function.

Execute a backward pass from right to left and spread the error to each node utilizing backpropagation. Estimate every weight donation to the mistake and change the weights appropriately using gradient descent.

An ANN is a set of related nodes provoked by the configuration and process of organic neurons in the brain. Learning a neural network input and output affects the information structure that flows through the network. Its major part is that the supervised learning technique can teach the method. During this procedure, the neural networks are needed to train a model utilizing detailed data, including a particular input and results that match the adding approach to be modeled [73].

Journal of Theoretical and Applied Information Technology
31st August 2022. Vol.100. No 16
© 2022 Little Lion Scientific

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4.17 Supervised Learning

Typically, MLPs are trained repetitively using back-propagation learning algorithms [74], updated through a network of weights and biases based on the propagation of output error F (xk) -yk using a gradient descent method. Thus, the network gradually learns from its error on each iteration.

Finally, for an ANN skilled to utilize inputs that are separate from the training instances, an independent testing set is needed to evaluate the generalization ability of the ANN model [75].

4.18 Ensemble Classifier

Ensemble classifiers stock the projections of numerous base approaches. Broadly practical and theoretical proof shows that model mixture raises predictive precision [76],[77]. Ensemble learners make the base systems in a dependent or independent way. For instance, the bagging algorithm Originated different base models from bootstrap samples of the primary data [78]. On the other hand, boosting algorithms increase an ensemble in a dependent trend. They Repetitively add a base approach that is qualified to ignore the mistakes of the existing ensemble [79]. The literature offers other additions to bagging and boosting [80]. The typical denominator of homogeneous ensembles is developing the base models utilizing the same classification algorithm (Figure 17) [81].

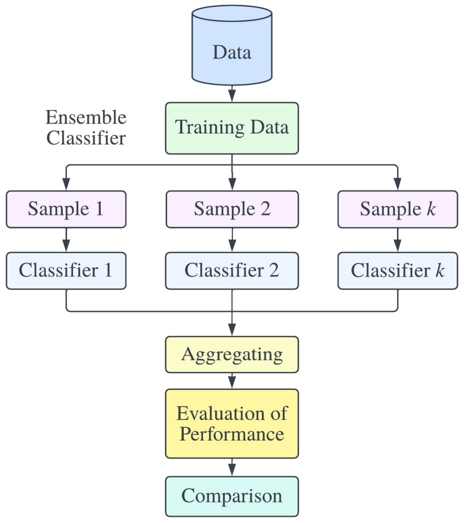

Figure 17: Workflow of single v. ensemble classifiers

4.19 Boosting

Boosting means meta-algorithm that can be seen as a model averaging approach. It is a widely utilized ensemble method and the most significant learning concept. Initially designed for classification, it can profitably expand this regression approach. The original boosting algorithm combined three weak learners [82].

4.20 Stacking

Stacking involves integrating multiple classifiers developed using various learning algorithms on a single dataset that structured pairs of feature vectors and their classifications. This approach is formed of primarily two steps; in the first step, a group of base-level classifiers is developed. In the second step, a meta-level classifier is understood, which mixes the results of the base-level classifiers [82].

4.21 Applications of Ensemble Methods

- Ensemble methods can be used as general diagnostic processes to develop a more traditional model. The more significant the difference in the standard quality between the ensemble method and a conventional model, the additional information is that the traditional model is perhaps absent.
- Ensemble methods can evaluate the connection between descriptive variables and the interest in traditional statistical models. Predictors or basis roles ignored in a usual model may appear with ensemble methods.
- It could better capture the selection process with the help of the ensemble method and estimate the probability of membership with a slight bias in each treatment group.
- One can utilize ensemble approaches to execute the covariance coordination inherent in various regression and connected processes. One will "residualized" the reply and the predictors of curiosity with ensemble methods.

Algorithm 1 Method for Ensemble Classifier

```
1:  Procedure ENSEMBLE (SVM, KNN, RF, NN)
2:  Load Trained models
3:  Compare the Performance of four classifiers
4:  Performing Majority Voting for every observation.
    Compare the performance of majority voting with all
    four Classifier
6:  Return Ensemble Classifier Accuracy
```

4.22 Vote

A vote is a meta-algorithm that uses different classifiers to complete the decision procedure [83]. It utilizes the strength of different individual classifiers and uses a mixed guide for decisions. For instance, majority voting, the lowest probability, a product of possibilities, highest probability, and average probabilities are various algorithms for combination guides. To deal with the multi-class classification, one could not choose majority voting cause the digit of classes is more than that of primary classifiers. This article uses the average of probabilities techniques to make a decision. The class label is chosen and established based on predicted probabilities' highest average value.

Think there is l classifiers E = {E1,..., l}, and e classes $\lambda = \{t_1, \ldots, t_e\}$, $l = 3$ set, the value of e rely on the attacks class. A classifier $E_1 : T^n \rightarrow [0, 1]^n$ receive an object $y \in T^n$ and outcomes a vector $[Q_{e_i}(t_1|y), \ldots Q_{e_i}(t_c|y)]$, all classes average probability is $Q_{e_i}(t_j|y)$, denotes the possibility by E_i, y belongs to class t_j every class t_j, let v_j presents the mean of the possibilities, imposed by the l classifiers, which can be measured as:

$$v_j = \frac{1}{l}\sum_{i=1}^{l} Q_{e_i}(t_j|y)$$

let $V = [v_1, \ldots v_e]$ be the set of mean probabilities for e classes. Then, y is imposed to the class t_k if v_k is the highest in V.

5. DEEP LEARNING BASE PROPOSED SYSTEM

Please provide a Graphical presentation of the difference between typical machine learning-based and deep learning-based methods. See example [84]. It is a graphical presentation of the distinction between a DL-based IDS and a classical ML-based IDS. Generally, the IDS has four main parts: data normalization, pre-processing, classification, and feature extraction (Figure 18). Further, Machine Learning techniques have evolved into a contentious issue for IDS. This is cause standard Machine Learning techniques to launch the data sparsity problem. Although classic Machine Learning techniques utilized a handmade feature extraction approach, it is tough to develop a helpful feature extraction method due to sample variation. The Deep Learning technique has achieved broad concern from researchers in different fields due to the ability of automated feature removal extract. Deep Learning is widely utilized in Cyber Security. Details in Figure 18.

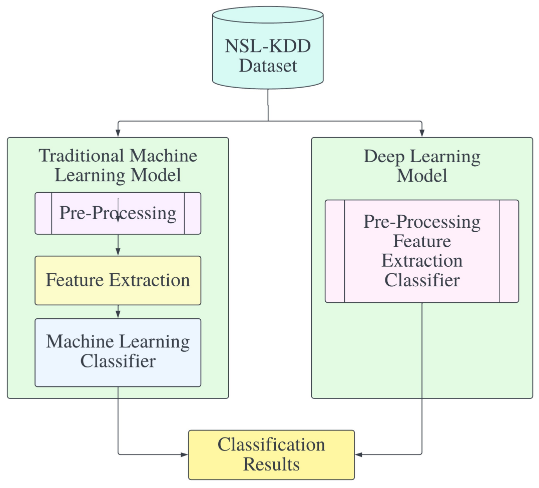
NSL-KDD Dataset
Traditional Machine Learning Model
Pre-Processing
Feature Extraction
Machine Learning Classifier
Deep Learning Model
Pre-Processing Feature Extraction Classifier
Classification Results

Figure 18: TML based IDS vs DL-based IDS.

Deep Learning needs high-end machines to converse with traditional MLA. Graphics Processing Unit (GPU) has now evolved into an integrated part to perform any Deep Learning Algorithm (DPA).

A significant advantage of deep learning, and a leading part in knowing why it will be famous, is that enormous quantities of data power it. The "Big Data Era" of technology will deliver vast opportunities for creativity in deep learning.

To reduce data complexity in traditional MLA, most used features need to be detected by a domain expert to create the patterns that are more visual for learning algorithms. The most significant benefit of DPA is that they incrementally attempt to know high-level elements from the information. This eradicates the necessity for domain expertise and critical feature taking out.

The significant difference between deep learning and MLA is the method of problem-solving. Deep Learning methods lean-to resolve the issue end to end, whereas Machine learning techniques require the problem statements to be split down into different features to be solved first, and then their effects to be mixed at the final step. For instance, for numerous object identification issues, Deep Learning approaches like Yolo net bring the image as input and deliver the position and name of objects at the result. But in standard MLA, such as SVM, bounding box Object Detection Algorithms (ODA) are first required to input Histogram of Oriented Gradient (HOGs) and specify all probable objects to accept relevant things.

Typically, a DPA brings a long time to teach due to many parameters. The famous ResNet algorithm needs about two weeks to teach entirely from scratch. On the other hand, Traditional MLA needs a few seconds to a few hours to prepare. In the testing phase, the scenario is completely reversed. The DPA requires little time to run at test time. On the other hand, if compared with K-nearest neighbors, the test time increases as the data size increases. Though it does not apply to all MLA, few have short testing times.

Interpretability is the central problem why numerous sectors utilize other ML techniques for Deep Learning. Let's look at an example.

There are used deep learning to estimate the related score of a document. Given performance is pretty excellent and close to human. But the problem is that it doesn't reveal why it gave that score. Mathematically it is possible to determine what points of a deep neural network were functioning well. Still, it is unknown which neurons were guessed to be a model and were doing it collectively. So, fail to explain the outputs, which are not in the subject of MLA, like logistic regression, decision trees, etc. [85].

Research has been conducted on the NSL-KDD dataset mentioned in [67], and MLA implemented on the dataset of NSL-KDD [9]. Regardless, the study emphasizes estimating the efficiency of the dataset by implementing MLA. The experimentation has been conducted using the Waikato Environment for Knowledge Analysis (WEKA) tool [86], and varieties of classifiers have been recorded based on that dataset's performance. The paper summarized that it is unnecessary to classify attacks to think about all the training features, and NSL-KDD is a revised edition of the Knowledge Discovery and Data Mining Tools Competition (KDD Cup'99) dataset [67]. Identified the same problem in [100], where noted analyzed standard ML algorithms and challenges to comparing different methods efficiency for the KDD CUP 99 dataset. The paper remarked that a significant issue in completing the comparative research is the shortage of a suitable method. The article recorded the typical cases with the KDD CUP 99 dataset. It proposed utilizing the perfect size of examples of a similar dataset to execute a comparative study of various attack classification approaches.

A set of instructions has been initiated in [87] to bridge the void between the existing needs of IDS datasets and their drawbacks. The article even discusses techniques to build datasets utilizing these instructions. Simulated datasets have been used in addition to the universally available datasets to measure the effectiveness of IDS on cable and wireless networks. [88], created a simulated dataset for mobile networks and applied ML algorithms to identify anomalies in wireless networks.

A hybrid method has been suggested, mixing the NB classifier with a feature-vitality-based decreasing approach [89]. The NSL-KDD performed simulations on the dataset and utilized a comparatively less set of features to classify the attack. The mentioned NSL-KDD dataset has 41 features. Twenty-five features have been reduced from here using the feature reduction approach. The precision of the suggested process has been shown to be 98%. The NSL-KDD dataset analysis used [90] unmanned ML system - K-means clustering. In this case, the 20% instance of the NSL-KDD dataset is divided into four clusters.

6. DATA PREPROCESSING

6.1 Normalization

There are non-numeric and 38 numeric classes in the NSL-KDD dataset. Since the input weights should be numeric, transform the non-numbered classes into numeric. For instance, the' protocol type' class has three features:' tcp',' udp', and' icmp'. Encode binary digits (0,0,1), (0,1,0) and (1,0,0). Transform the 41 into a 122-dimensional class map using this approach. There are different characteristics in the dataset in which the contrast between the max and min values is enormous. Such features are dst_bytes [0,1.3 × 109], src_bytes [0,1.3 ×109] and duration [0,58329]. We use the logarithmic scaling method to reduce the contrasts and then utilize the formula below to map it to the [0,1] span: xi = (xi - Min) / (Max-Min).

6.2 Convolutional Neural Network (CNN)

Convolution and Pooling are two operations in CNN. A set of convolution kernels or filters is utilized to convert input data to output. Input data properties display the result produced. So, the output is called a feature map. The running function works to process the convolution output further, and down-sampling Pooling is used to close irrelevant data. Pooling removes any errors in the data. In this way, learning is improved for the levels. Adjusting kernels/filters is used after learning CNN rounds so that class maps can effectively present input data.

The data goes to the input layer in a classical neural network; then, it moves to a hidden output layer and the output layer. All the layers are linked, and there is no connection in the similar layer between nodes.

TNN has many problems that cannot solve. CNN's architecture is much better than SNN's. Image classification has achieved remarkable results due to CNN in cases like speech analysis. CNN includes:

- One or more convolutional layers

- Pooling layers at the top
- Fully connected layers
- Dropout layers serve as regularization layers.

For structural reasons, CNN can take advantage of 2D framework input data. Could take a picture such as a network input. Thus, we bypass classical recognition algorithms' complex feature extraction and excessive data construction. The proficiency of modeling can be grown, and the disadvantage of the manual data processing approach can be reduced by transferred weights, sparse connectivity, and pooling. CNN can acquire knowledge from different classes of attributes from a significant quantity of unlabeled data. Thus, how can utilize CNN for network intrusion detection is very wide. Figure 19 shows the architecture of a system similar to the suggested.

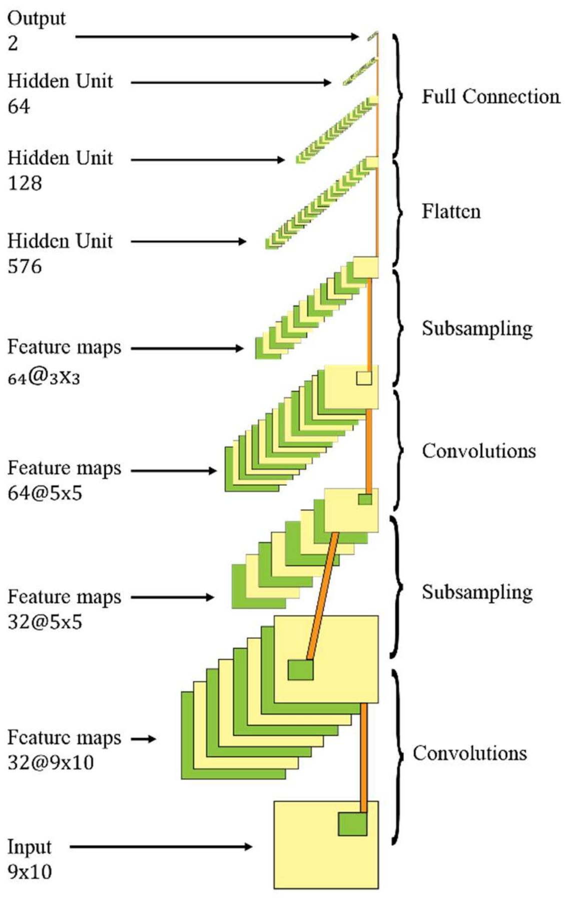

Figure 19: The architecture of a single CNN

7. EVALUATION METRICS

There are used Accuracy (AC) to calculate the performance of the method. Therefore, it also initiates FPR and identification rate. TP denotes the number of data that remove perfectly and detected as anomalies. Thus, TN represents the inverse. TN means the appropriate common data, and FN denotes the inverse.

We follow the notations given below: Accuracy, Accuracy = (True Positive + True Negative) / (True Positive + True Negative + False Positive + False Negative) True Positive Rate, True Positive Rate = True Positive / (True Positive + False Negative) False Positive Rate = False Positive / (False Positive + True Negative) True Negative Rate = True Negative / (True Negative + False Positive) False Negative Rate = False Negative / (False Negative + True Positive) Sensitivity, True Positive Rate = True Positive / (True Positive + False Negative) Specificity, False Positive Rate = True Negative / (True Negative + False Positive) Therefore, our motivation is to get high accuracy and better detection rate with low false positive.

On the entire group of data, the CNN paused training at era 45, getting a precision of 97.24%. The activity defeat in the last era came at 0.516, while the verification defeat came at 0.7915. The approach precision overages are displayed in Figure 20, while the training and confirmation losses for every era are shown in Figure 21.

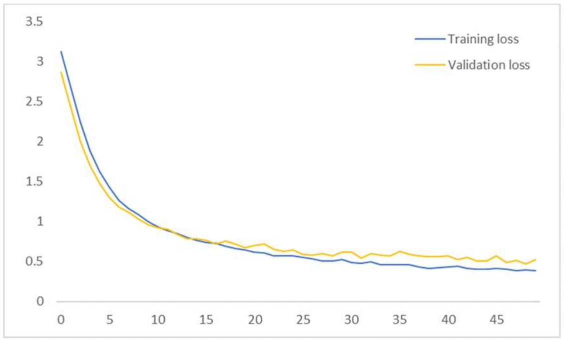

Figure 20: Training loss vs. Validation loss.

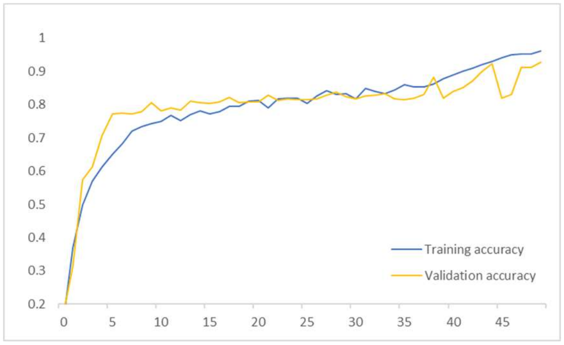

Figure 21: Training Accuracy vs. Validation Accuracy.

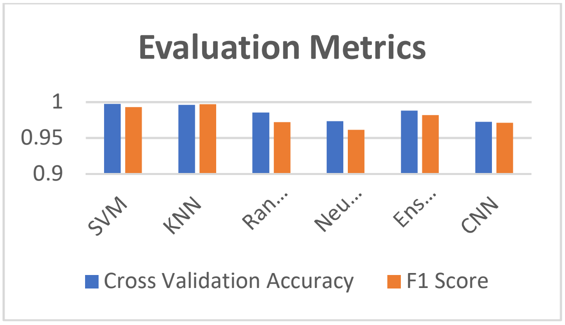

Figure 22: Bar plots of CVA and F1 Score characteristic values for various classification models, including SVM, KNN, RC, ANN, EC, and CNN.

8. EXPERIMENT RESULTS AND DISCUSSIONS

In this part, the effects of the presented ensemble models are studied and performed further analysis to gain results. In the first investigation, training was completed utilizing all approaches. A functional IDS should attain high precision, recall, and F1-measure with low FAR. We also estimate the accuracy, precision, recall, and F1 measure for each classifier, where precision is evaluated by

$$\text{Accuracy} = \frac{\text{TP+}}{\text{TP+TF+FP+F}}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP+}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}$$

$$\text{F1} = \frac{2*\text{Precision}*\text{Recall}}{\text{Precision+Re}}$$

TPs are the quantity of attack data that is ideally classified; TNs are the quantity of common visitor data; FP is the falsely classified quantity of common visitor data. FN is the quantity of attack data instances falsely classified. Table 7 lists the last three tests of IDS performance evaluation. From this study, it can be thought that the entire network features may make some small contributions, some may not contribute, some may not contribute equally, and some may contribute more to the classification strategy. Thus, creating the IDS based on the entire traditional NCF outputs reduces detection precision while growing the FARs, the analysis time, and the difficulty of the utilized IDS. All these earlier cons are not appropriate entirely in the IDSs. Secondly, optimizing the ID approach through adjusting its parameters outputs in enhancing the IDS's proficiency by growing the classification precision, detection ratio, True Positive Rate, and True Negative Rate while reducing the identification time and False Positive Rate along with False Negative Rate.

Table 6: Model accuracy in binary classification setting

Model	Cross Validation Accuracy			
	DoS	U2R	R2L	Probe
SVM	0.99715	0.99632	0.96793	0.9845
KNN	0.9962	0.99703	0.96737	0.99077
Random Classifier	0.98512	0.94123	0.96907	0.9523
Neural Network	0.97322	0.9754	0.95127	0.9812
Ensemble Classifier	0.98792	0.97749	0.96391	0.97719

Table 7: Model F1 score in binary classification setting

Model	F1 Score			
	DoS	U2R	R2L	Probe
SVM	0.99278	0.84869	0.95529	0.97613
KNN	0.99672	0.87831	0.95389	0.98553
Random Classifier	0.9719	0.94105	0.96129	0.9872
Neural Network	0.9611	0.98301	0.9863	0.9791
Ensemble Classifier	0.9819	0.98512	0.98512	0.9895

CONCLUSION

This paper proposed an effective IDS to identify various attacks (Probe, Dos, U2R, R2L) utilizing an FSS connected with the SVM, KNN, RF, NN, and ECA to develop a robust intrusion detection method. The offered IDS is qualified and experimented on the standard NSL-KDD data source. The valuation outcomes expressed its effectiveness in acknowledging the expected behaviors and identifying the attacks with the best detection precision and small ratio of false alarms.

The coming tasks are guided on utilizing other evolutionary strategies to optimize the algorithm's managing features and other perfect NFS techniques.

REFERENCES:

[1] Debar, H., Dacier, M., and Wespi, A., "Towards a taxonomy of intrusion- detection systems," *Computer networks,* vol. 31, no. 8, 1999, pp. 805–822.

[2] Garcia-Teodoro, P., Diaz-Verdejo, J., Maci a´-Ferna´ndez, G., and Va´zquez, E., "Anomaly-based network intrusion detection: Techniques, systems and challenges," computers & security, vol. 28, no. 1-2, 2009, pp. 18– 28. https://doi.org/10.1016/j.cose.2008.08.003

[3] Moradi, M. and Zulkernine, M., "A neural network-based system for intrusion detection and classification of attacks," *In Proceedings of the IEEE international conference on advances in intelligent systems theory and applications*. IEEE Lux-embourg-Kirchberg, Luxembourg, 2004, pp. 15–18. https://people.ece.ubc.ca/moradi/148-04-MM-MZ.pdf

[4] Mukkamala, S., Sung, A. H., and Abraham, A., "Intrusion detection using an ensemble of intelligent paradigms," *Journal of network and computer applications*, vol. 28, no. 2, pp. 167–182, 2005, http://dx.doi.org/10.1016/j.jnca.2004.01.003

[5] Xue, J.-S., Sun, J.-Z., and Zhang, X., "Recurrent network in network intrusion detection system," *In Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 04EX826)*, vol. 5. IEEE, pp. 2676–2679, 2004, http://dx.doi.org/10.1109/ICMLC.2004.1378292

[6] Kevric, J., Jukic, S., and Subasi, A., "An effective combining classifier approach using tree algorithms for network intrusion detection," *Neural Computing and Applications*, vol. 28, no. 1, pp. 1051–1058, 2017, https://link.springer.com/article/10.1007/s00521-016-2418-1

[7] Mehibs, S. M. and Hashim, S. H., "Proposed network intrusion detection system based on fuzzy c mean algorithm in cloud computing environment," *Journal of University of Babylon for Pure and Applied Sciences*, vol. 26, no. 2, 2018, pp. 27–35, http://dx.doi.org/10.29196/jub.v26i2.471

[8] Pham, N. T., Foo, E., Suriadi, S., Jeffrey, H., and Lahza, H. F. M., "Improving performance of intrusion detection system using ensemble methods and feature selection," *In Proceedings of the Australasian Com- puter Science Week Multiconference*, pp. 1–6, 2018, http://dx.doi.org/10.1145/3167918.3167951

[9] Tavallaee, M., Bagheri, E., Lu, W., and Ghorbani, A. A., "A detailed analysis of the kdd cup 99 data set," in *2009 IEEE symposium on computational intelligence for security and defense applications*. IEEE, pp. 1–6, 2009, https://doi.org/10.1109/CISDA.2009.5356528

[10] Ferentinos, K. P., "Deep learning models for plant disease detection and diagnosis," *Computers and Electronics in Agriculture*, vol. 145, pp. 311– 318, 2018. https://doi.org/10.1016/j.compag.2018.01.009

[11] Aarti Kumthekar, R. R. G., "Ensemble learning technique for cloud classification," *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 9, pp. 2582–2587, 2019. doi: 10.35940/ijeat.B3957.129219

[11] Wooseok Seo, Wooguil Pak, "Real-Time Network Intrusion Prevention System Based on Hybrid Machine Learning", *IEEE Access,* pp. (99):1-1, March 2021. https://doi.org/10.1109/ACCESS.2021.3066620

[12] Al-Jarrah, O.Y., Alhussein, O., Yoo, P.D., Muhaidat, S., Taha, K., and Kim, K., "Data randomization and cluster-based partitioning for botnet intrusion detection," *IEEE transactions on cybernetics*, vol. 46, pp. 1796–1806. 2015, doi:10.1109/TCYB.2015.2490802.

[13] Elhag, S., Fernández, A., Bawakid, A., Alshomrani, S., and Herrera, F., "On the combination of genetic fuzzy systems and pairwise learning for improving detection rates on intrusion detection systems," *Expert Systems with Applications,* vol. 42, pp. 193–202, 2014, doi:10.1016/j.eswa. 2014.08.002.

[14] Wang, K., Du, M., Maharjan, S., and Sun, Y., "Strategic honeypot game model for distributed denial of service attacks in the smart grid," *IEEE Transactions on Smart Grid*, vol. 8, pp. 2474–2482, 2017, doi:10.1109/TSG. 2017.2670144.

[15] Wang, K., Du, M., Sun, Y., Vinel, A., and Zhang, Y., "Attack detection and distributed forensics in machine-to-machine networks," *IEEE Network, vol.* 30, pp. 49–55, 2016a, doi:10.1109/MNET.2016.1600113NM.

[16] Wang, K., Du, M., Yang, D., Zhu, C., Shen, J., and Zhang, Y., "Game-theory-based active defense for intrusion detection in cyberphysical embedded systems," *ACM Transactions on Embedded Computing Systems (TECS),* vol. 16, No. 18, 2016b, doi:10.1145/2886100.

[17] Joldzic, O., Djuric, Z., Vuletic, P., "A transparent and scalable anomaly-based dos detection method," *Computer Networks*, vol. no. 104, pp. 27–42, 2016, doi:10.1016/j.comnet.2016.05.004.

[18] Papamartzivanos, D., Mármol, F.G., Kambourakis, G., "Dendron: Genetic trees driven rule induction for network intrusion detection systems," *Future Generation Computer Systems*, vol. 79, pp. 558–574, 2018, doi:10.1016/j.future.2017.09.056.

[19] Kotsiantis, S., Kanellopoulos, D., and Pintelas, P., "Data preprocessing for supervised leaning," *International Journal of Computer Science*, vol. 1, pp. 111–117, 2018, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.104.8413&rep=rep1&type=pdf

[20] Du, M., Wang, K., Chen, Y., Wang, X., and Sun, Y., "Big data privacy preserving in multi-access edge computing for heterogeneous internet of things" *IEEE Communications Magazine*, vol. 56, pp. 62–67, 2018a, doi:10.1109/MCOM.2018.1701148

[21] Du, M., Wang, K., Xia, Z., and, Zhang, Y., "Differential privacy preserving of training model in wireless big data with edge computing," *IEEE Transactions on Big Data*, 2018b, doi:10.1109/TBDATA.2018.2829886

[22] Mishra, P., Varadharajan, V., Tupakula, U., and Pilli, E.S., "A detailed investigation and analysis of using machine learning techniques for intrusion detection," *IEEE Communications Surveys & Tutorials,* 2018, doi:10.1109/COMST.2018.2847722.

[23] Feng, X., Xiao, Z., Zhong, B., Qiu, J., and Dong, Y., "Dynamic ensemble classification for credit scoring using soft probability," *Applied Soft Computing* 65, 139–151, 2018 doi:10.1016/j.asoc.2018.01.021.

[24] Salo, F., Nassif, A.B., Essex, A., "Dimensionality reduction with ig-pca and ensemble classifier for network intrusion detection," *Computer Networks*, vol. 148, pp. 164–175, 2019, doi:10.1016/j.comnet.2018.11.010.

[25] Pham, N.T., Foo, E., Suriadi, S., Jeffrey, H., and Lahza, H.F.M., "Improving performance of intrusion detection system using ensemble methods and feature selection," *In Proceedings of the Australasian Computer Science Week Multiconference, ACM.,* p. 2, 2018, doi:10.1145/3167918.3167951.

[26] Aljawarneh, S., Aldwairi, M., Yassein, M.B., Anomalybased intrusion detection system through feature selection analysis and building hybrid efficient model. *Journal of Computational Science*, vol. 25, pp. 152–160, 2009, doi:10.1016/j.jocs.2017.03.006.

[27] Hota, H., and Shrivas, A.K., 2014. Decision tree techniques applied on nsl-kdd data and its comparison with various feature selection techniques, *In Proceeding Advanced Computing, Networking and Informatics,* vol. 1, Springer, pp. 205–211. doi:10.1007/978-3-319-07353-8_24

[28] Khammassi, C., and Krichen, S., "A galr wrapper approach for feature selection in network intrusion detection." *Computers & security,* vol. 70, pp. 255–277, 2017, doi:10.1016/j.cose.2017.06.005.

[29] Yuyang Zhou, Guang Cheng, Shanqing Jiang, and Mian Dai," Building an Efficient Intrusion Detection System Based on Feature Selection and Ensemble Classifier," *Computer Networks* vol. 174, p. 2, June 2020, https://www.sciencedirect.com/science/article/abs/pii/S1389128619314203#:~:text=https%3A//doi.org/10.1016/j.comnet.2020.107247

[30] Roesch M., "Snort - Lightweight Intrusion Detection for Networks," Proc. of LISA '99 Proceedings of the 13th USENIX conference on System, Vol. 99, No. 1, pp. 229-238, November, 1999.

[31] Sedgewick R., "Algorithms in C: Parts 1-4, Fundamentals, Data Structures, Sorting, and Searching, Boston (USA)," *Addison-Wesley Longman Publishing,* 3rd ed., p. 702, 1997.

[32] Wang W., Sheng Y., Wang J., Zeng X., Ye X., Huang Y., and Zhu M., "HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection," *IEEE Access,* vol. 6, pp. 1792-1806, December, 2017, https://doi.org/10.1109/ACCESS.2017.2780250

[33] Hu Q., Asghar M. R., Brownlee N., "Evaluating network intrusion detection systems for high-speed networks," Proc. of Telecommunication Networks and Applications Conference (ITNAC), pp. 1-6, November, 2017, https://doi.org/10.1109/ATNAC.2017.8215374

[34] Ertam F., Yaman O., "Intrusion detection in computer networks via machine learning algorithms," *Proc. of 2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*, pp. 1-4, September, 2017, https://doi.org/10.1109/IDAP.2017.8090165

[35] Tang T. A., Mhamdi L., McLernon D., Zaidi S. A. R., and Ghogho M., "Deep learning approach for Network Intrusion Detection in Software Defined Networking," In *proceeding of 2016 International Conference on Wireless Networks*

and Mobile Communications (WINCOM), pp. 1-6, October, 2016, https://doi.org/10.1109/WINCOM.2016.7777224

[36] Zhang J., Zulkernine M., and Haque A., "Random-forests-based network intrusion detection systems," In Proceeding of IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 38, no. 5, pp. 649-659, August, 2008, https://doi.org/10.1109/TSMCC.2008.923876

[37] Breiman L., "Random Forests," *Machine learning*, vol. 45, no. 1, pp. 5-32. October, 2001, https://link.springer.com/article/10.1023/a:1010933404324

[38] Wattanapongsakorn N., Srakaew S., Wonghirunsombat E., Sribavonmongkol C., Junhom T., Jongsubsook P., and Charnsripinyo C., "A Practical Network-Based Intrusion Detection and Prevention System," *In Proceeding of 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications,* pp. 209-214, June, 2012, https://doi.org/10.1109/TrustCom.2012.46

[39] Ahmad I., Basheri M., Iqbal MJ., and Rahim A., "Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection," *IEEE Access,* vol. 6., pp. 33789-33795, May, 2018, https://doi.org/10.1109/ACCESS.2018.2841987

[40] Sahu S., and Mehtre BM., "Network intrusion detection system using J48 Decision Tree," *In Proceeding of International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 2023-2026, August, 2015, https://doi.org/10.1109/ICACCI.2015.7275914

[41] Cheong Y., Park K., Kim H., Kim J., and Hyun S., "Machine Learning Based Intrusion Detection Systems for Class Imbalanced Datasets," *Journal of the Korea Institute of Information Security & Cryptology,* vol. 27, no.-6, pp. 1385-1395, December, 2017, https://doi.org/10.13089/JKIISC.2017.27.6.1385

[42] Yin C., Zhu Y., Fei J., He X., "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks," *IEEE Access,* Vol. 5, pp. 21954-21961, November, 2017, https://doi.org/10.1109/ACCESS.2017.2762418

[43] Park K., Song Y., Cheong Y., "Classification of Attack Types for Intrusion Detection Systems Using a Machine Learning Algorithm," *In proceeding of 2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigDataService),* pp. 282-286, July, 2018, https://doi.org/10.1109/BigDataService.2018.00050

[44] Lin W., Lin H., Wang P., Wu B., and Tsai J., "Using convolutional neural networks to network intrusion detection for cyber threats," Proc. of 2018 IEEE International Conference on Applied System Invention (ICASI), pp. 1107-1110, June, 2018, https://doi.org/10.1109/ICASI.2018.8394474

[45] Al-Qatf M., Lasheng Y., Al-Habib M., and Al-Sabahi K., "Deep Learning Approach Combining Sparse Autoencoder with SVM for Network Intrusion Detection," *IEEE Access,* vol. 6, pp. 52843- 52856, September, 2018, https://doi.org/10.1109/ACCESS.2018.2869577

[46] Soheily-Khah S., Marteau P., and Béchet N., "Intrusion Detection in Network Systems Through Hybrid Supervised and Unsupervised Machine Learning Process: A Case Study on the ISCX Dataset," *In Proceeding of the 1st International Conference on Data Intelligence and Security (ICDIS),* pp. 219-226, May, 2018, https://doi.org/10.1109/ICDIS.2018.00043

[47] Yuan Y., Huo L., and Hogrefe D., "Two Layers Multi-class Detection method for network Intrusion Detection System," *In proceeding of IEEE Symposium on Computers and Communications (ISCC),* pp. 767- 772., July, 2017, http://dx.doi.org/10.1109/ISCC.2017.8024620

[48] Abdullah, M., Balamash, A., Alshannaq, A., and Almabdy, S., "Enhanced intrusion detection system using feature selection method and ensemble learning algorithms," *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 16, 2018.

[49] Gaikwad, D., and Thool, R.C., "Intrusion detection system using bagging ensemble method of machine learning," *In proceeding of 2015 International Conference on Computing Communication Control and Automation,* pp. 291–295, 2015, doi:10.1109/ICCUBEA.2015.61.

[50] Jabbar, M., Aluvalu, R., Reddy, S.S.S., "Cluster based ensemble classification for intrusion detection system," *In proceeding of the 9th International Conference on Machine Learning and Computing,* pp. 253–257, 2017, doi:10.1145/3055635.3056595.

[51] Paulauskas, N., Auskalnis, J., "Analysis of data pre-processing influence on intrusion detection using nsl-kdd dataset," *In proceeding of 2017 Open Conference of Electrical, Electronic and Information Sciences (eStream),* pp. 1–5, 2017, doi:10.1109/eStream.2017.7950325.

[52] Moustafa, N., Turnbull, B., Choo, K.K.R., "An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things," *IEEE Internet of Things Journal*, 2018, doi:10.1109/JIOT.2018.2871719.

[53] Malik, A.J., Shahzad, W., Khan, F.A., "Network intrusion detection using hybrid binary pso and random forests algorithm," *Security and Communication Networks,* vol. 8, no. 16, 2012, pp. 2646–2660. doi:10.1002/sec.508.

[54] Zhong, Y., Chen, W., Wang, Z., Chen, Y., Wang, K., Li, Y., Yin, X., Shi, X., and Yang, J., Li, K., "Helad: A novel network anomaly detection model based on heterogeneous ensemble learning," *Computer Networks*, vol. 169, no. 107049, 2020, doi:10.1016/j.comnet.2019.107049.

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

[56] Shadi Aljawarneh, Monther Aldwairi, and Muneer Bani Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," *Journal of Computational Science,* vol. 25, pp. 152-160, 2018, https://doi.org/10.1016/j.jocs.2017.03.006

[57] Kunal, and Mohit Dua, "Attribute Selection and Ensemble Classifier based Novel Approach to Intrusion Detection System," *International Conference on Computational Intelligence and Data Science (ICCIDS),* vol. 167, 2020, pp. 2191-2199, https://doi.org/10.1016/j.procs.2020.03.271

[58] Catal, C., and Nangir, M., "A sentiment classification model based on multiple classifiers," *Applied Soft Computing*, vol. 50, pp. 135–141., 2017, doi:10. 1016/j.asoc.2016.11.022.

[59] Catillo, M., Rak, M., and Villano, U., "Discovery of dos attacks by the zed-ids anomaly detector," *Journal of High Speed Networks*, pp. 1–17, 2019, doi:10.3233/JHS-190620.

[60] Hajisalem, V., and Babaie, S., "A hybrid intrusion detection system based on abc-afs algorithm for misuse and anomaly detection," *Computer Networks*, vol. 136, pp. 37–50, 2018, doi:10.1016/j.comnet.2018.02.028.

[61] Bolón-Canedo, V., Sánchez-Maroño, N., and Alonso-Betanzos, A., "Feature selection for high-dimensional data," *In proceeding of Artificial Intelligence*, vol. 5, pp. 65–75, 2016, doi:10.1007/s13748-015-0080-y.

[62] Liu, H., and Yu, L., "Toward integrating feature selection algorithms for classification and clustering," *IEEE Transactions on Knowledge & Data Engineering*, pp. 491–502, 2005, doi:10.1109/TKDE.2005.66.

[63] S.Subash Chandra Bose, T. Christopher, "Aggregate Linear Discriminate Analyzed Feature Extraction and Ensemble of Bootstrap with Knn Classifier for Malicious Tumour Detection," *International Journal of Recent Technology and Engineering (IJRTE),* vol. 8, no.3, September 2019, https://www.ijrte.org/wp-content/uploads/papers/v8i3/C4802098319.pdf

[64] Lee, W., Stolfo, S.J., and Mok, K.W., "A data mining framework for building intrusion detection models," *In proceedings of the 1999 IEEE Symposium on Security and Privacy*, IEEE, pp. 120–132, 1999, doi:10.1109/SECPRI.1999.766909.

[65] Khan, M.A., Karim, M., and Kim, Y., et al., "A scalable and hybrid intrusion detection system based on the convolutional lstm network," *Symmetry*, vol. 11, no. 583, 2019, doi:10.3390/sym11040583.

[66] Bala, R., and Nagpal, R., "A review on kdd cup99 and nsl nsl-kdd dataset," *International Journal of Advanced Research in Computer Science 10.* 2019, https://doi.org/10.26483/ijarcs.v10i2.6395

[67] L. Dhanabal and S. P. Shantharajah, "A study on NSL-KDD dataset for intrusion detection system based on classification algorithms," *International Journal of Advanced Research in Computer and Communication Engineering,*

vol. 4, no. 6, pp. 446-452, 2015, DOI: 10.17148/IJARCCE.2015.4696

[68] Prwatech, "Support Vector Machine Tutorial for Beginners," *Prwatech,* 2020, https://prwatech.in/blog/machine-learning/support-vector-machine-tutorial-for-beginners.

[69] Tavish, "Introduction to k-Nearest Neighbors: A powerful Machine Learning Algorithm (with implementation in Python & R). *Analytics Vidhya,* 2018, https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering

[70] Onesmus Mbaabu, "Introduction to Random Forest in Machine Learning," *Section,* 2020, https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning

[71] Feng, Q., Liu, J., and Gong, J., "Uav remote sensing for urban vegetation mapping using random forest and texture analysis," *Remote sensing,* vol. 7, pp. 1074–1094, 2015, doi:10.3390/rs70101074

[72] Afroz Chakure, "Random Forest Classification and its implementation in Python," *The Startup,* 2019, https://medium.com/swlh/random-forest-classification-and-its-implementation-d5d840dbead0

[73] Ivan Argatov, "Artificial Neural Networks (ANNs) as a Novel Modeling Technique in Tribology," 2019 https://doi.org/10.3389/fmech.2019.00030

[74] Rumelhart, D., Hinton, G., and Williams, R. "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986, doi: 10.1038/323533a0

[75] Bishop, C. "Neural Networks for Pattern Recognition," *Oxford: Oxford University Press,* 1995

[76] Finlay, S., "Multiple classifier architectures and their application to credit risk assessment," *European Journal of Operational Research,* vol. 210, pp. 368-378, 2011, https://doi.org/10.1016/j.ejor.2010.09.029

[77] Paleologo, G., Elisseeff, A., & Antonini, G. "Subagging for credit scoring models," *European Journal of Operational Research,* vol. 201, pp. 490-499, 2010, https://doi.org/10.1016/j.ejor.2009.03.008

[78] Breiman, L., "Bagging predictors," *Machine Learning,* vol. 24, pp. 123-140, 1996

[79] Freund, Y., & Schapire, R. E., "Experiments with a New Boosting Algorithm," *In proceeding of the 13th Intern. Conf. on Machine Learning*, pp. 148-156, 1996, https://cseweb.ucsd.edu/~yfreund/papers/boostingexperiments.pdf

[80] Rodriguez, J. J., Kuncheva, L. I., & Alonso, C. J., "Rotation forest: A new classifier ensemble method," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 28, pp. 619-1630, 2006, https://doi.org/10.1109/TPAMI.2006.211

[81] Lessmann, S., Baesens, B., Seow, HV and Thomas, LC., "Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research," *European Journal of Operational Research,* vol. 247, no. 1, pp. 124-136, 2015, https://doi.org/10.1016/j.ejor.2015.05.030

[82] Ambika Choudhury, "Basics of Ensemble Learning in Classification Techniques Explained," Analytics India Mag, 2019 https://analyticsindiamag.com/basics-of-ensemble-learning-in-classification-techniques-explained.

[83] Catal, C., and Nangir, M., "A sentiment classification model based on multiple classifiers," *Applied Soft Computing,* vol. 50, pp. 135–141, 2017, doi:10.1016/j.asoc.2016.11.022

[84] Sharaf J. Malebary, and Arshad Hashmi, "Automated Breast Mass Classification System Using Deep Learning and Ensemble Learning in Digital Mammogram," *IEEE Access,* vol. 9, pp. 55312-55328, 2021, https://doi.org/10.1109/ACCESS.2021.3071297

[85] Sambit Mahapatra, "Why Deep Learning over Traditional Machine Learning?" *Towards Data Science,* 2018, https://towardsdatascience.com/why-deep-learning-is-needed-over-traditional-machine-learning-1b6a99177063

[86] Srivastava S., "Weka: a tool for data preprocessing, classification, ensemble, clustering and association rule mining," *International Journal of Computer Applications,* vol. 88, no. 10, 2014, http://dx.doi.org/10.5120/15389-3809

[87] Shiravi A, Shiravi H, Tavallaee M, and Ghorbani AA, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Computers & security,* vol. 31, no. 3, pp. 357–374, 2012, https://doi.org/10.1016/j.cose.2011.12.012

[88] Damopoulos D, Menesidou SA, Kambourakis G, Papadaki M, Clarke N, and Gritzalis S., "Evaluation of anomaly-based IDS for mobile devices using machine learning classifiers," *Security and Communication Networks,* vol. 5, no. 1, pp. 3–14, 2012, http://dx.doi.org/10.1002/sec.341

[89] Mukherjee S, and Sharma N., "Intrusion detection using naive Bayes classifier with feature reduction," *Procedia Technology,* vol. 4, pp. 119–128, 2012, https://doi.org/10.1016/j.protcy.2012.05.017

[90] Kumar V, Chauhan H, and Panwar D., "K-means clustering approach to analyze NSL-KDD intrusion detection dataset," *International Journal of Soft Computing and Engineering (IJSCE),* 201