UtilitasMathematica
ISSN 0315-3681 Volume 122 (2), 2025

An Intrusion Detection System for Preserving Information Security in Cloud Environment

[1]Ashima Jain, [2]Ashima Narang, [3]Manju*
[1, 2]Department of Computer Science and Engineering, Amity University, Gurugram, Haryana, India.
[3]Department of Computer Science and Engineering, PES University, Bangalore, India
[1]ashimajain046@gmail.com, [2]Ashimanarang04@gmail.com, *[3]manju.nunia@gmail.com

Abstract:

Cloud computing acquires additional security risks needing Intrusion Detection Systems (IDS) which aid in confirming the safety and dependability of systems and networks in the cloud environment. Increasing cloud security is the focus of this paper which proposes a framework of The Probabilistic Optimized Feature Voting Classification (POFVC) which aims to strengthen intrusion detection systems mechanisms in cloud computing environments. In this regard, the paper proposed the POFVC model which employs probabilistic optimization and feature voting to enhance the precision and production of intrusion detection. POFVC integrates machine learning techniques to fight the feature selection challenge at a progressive level, showing great outcomes on datasets widely known for intrusion detection such as UNSW-NB15, KDD Cup 1999, NSL-KDD, CICIDS2017, AWID, and DARPA Intrusion Detection Data. With accuracy greater than 98%, POFVC outshines conventional models SVM and Logistics, as well as other further evaluation metrics.
Keywords: Intrusion Detection System (IDS), Cloud Computing, Optimization, Probabilistic Model, Information Security

1. Introduction

The cloud environment has as of late metamorphosed into a critical aspect of the technology ecosystem. Business enterprises are quickly embracing multi-clouds, which are models of using the power of multiple cloud service providers to ensure optimum efficiency [1]. The introduction of edge computing has addressed the latency issues and facilitated processing of data in real-time at the edge and has thus revolutionized IoT and autonomous cars. The serverless computing simplified the growth of applications through containerization and Kubernetes orchestration in clouds environments [2]. Numerous cloud providers are continuously working on AI and their services which enhanced the provision of advanced analytics services. Besides that, hybrid and multi-cloud management tools have been created to support the smooth organization of workloads [3]. Using serverless computing, powerful databases can assist businesses to have a scalable data management service with the quantum computing services implying a new era of unprecedented computing power [4].

Distribution of services and data in a cloud environment has not only attracted authentic users, but has also introduced cyber attackers who are occasionally trying to use the surrounding to their own advantage [5]. Various attacks and unauthorized attempts are becoming more and more on a daily basis. In literature, the so called DDoS attacks are attacks used to deny access to a service by overloading it through traffic [6]. In addition, data breaches are also quite a popular form of attacks where sensitive information is stolen without authorization as a result of poor controls in the mis-configuration of the system security. Malware has also made its way to cloud environments infecting virtual machines and holding the data hostage at ransom for decryption [7]. Phishing is also a threat attacking users of the cloud by way of emails or fake login pages just to take hold of credentials. Threats from inside the organization, whether done consciously or subconsciously incurs a cost in terms of data availability within data integrity [8].Providers and cloud users must adopt active policies followed by strict security protocols for maintaining the integrity, availability, and confidentiality which include revision of unprotected system settings, the employment of sensitive information encryption, access control, intrusion suppression technologies, and abnormal activity detection systems [9].Malicious activities have the potential to be detected and mitigated through the use of IDS. As per the given research works, the IDS is basically known for as the security tool that searches and use system behaviours to devise a trend that could serve as an indicator of an attack [10]. An IDS takes on even greater significance in the world of cloud computing, where data and applications reside in virtualized environments on dispersed remote servers. A cloud based IDS works by monitoring the network traffic, system logs and application behaviours in the cloud framework over the period of time [11]. It uses refined algorithms and signature based detection methods which look at the patterns observed in relation to a known attack signature to limit the presence of wide spread threats such as DDoS, malware and unauthorized access [12]. Moreover, cloud IDS is able to recognize noticeable deviations from the set baselines which are often revealing of unknown threats [13]. These may consist of anomalous data retrieval, disproportionately high resource usage, and abnormal logins. The moment such deviations are recognized, the IDS

will set off alerts or some automated actions that allows administering the steps towards gaining control of the situation [14].

Integrating other security measures like firewalls and intrusion prevention systems (IPS) often form an amalgamated defence umbrella against malicious activities which is often referred to as cloud IDS solution [15]. Claiming from a cloud customer perspective, actively monitoring and analysing cloud data and behaviours is also a way to bolster the security stance of cloud environments supporting organizations to proactively mitigate potential threats looking after their information and services in the cloud [16]. A cloud-based IDS relies primarily on in-depth recognition of attack patterns or signature-based detection. New incoming network traffic, system logs, and application actions are verified for parameters with incoming attack signature databases to check for possible matches [17]. Such parameters are known pieces of data that associate attacks such as viruses, malware along with hacking tactics, and numerous others. Nevertheless, the ever-changing cloud setting continuously shifts the horizon of fresh threats and dangers that persistently crop up [18]. Automated isolation of anomaly uses a different approach. Remove all preconceived notions and set attack signatures [19].Also, cloud intelligence and defence systems (IDS) are crafted to ensure real-time data and activity monitoring within the client's environments. These systems keep track of the user's activities and how the resources are being utilized as well as data traffic and access [20].

The contribution of the proposed Probabilistic Optimized Feature Voting Classification (POFVC) is stated as the follows:

1. The paper presents a new technique of feature selection that is based on optimization and probabilistic heuristics. It focuses on enhancing the model by feature selection, otherwise known as feature reduction, which improves the model's efficiency.
2. POFVC improves classification at the feature level (ETS) by optimized feature selection compared with the previous approaches for classifying attacks. Better intrusion detection is achieved at the classification layer for different datasets.
3. The proposed model demonstrates scalability and adaptability where the model is described to perform well with the datasets such as UNSW-NB15, KDD Cup 1999, NSL-KDD, CICIDS2017, AWID (AIS-HIDS), and DARPA Intrusion Detection Data. Thus, the model can be recognized as a consistent capable tool for network security various scenarios.
4. The paper includes rule-based optimization which permits precision optimization to the classification rules set.

The paper's contributes in introducing a dominant and adaptable classification model for intrusion detection, backed by innovative feature selection techniques and rule-based optimization, finally enhancing network security in various real-world scenarios.

2. Literature Survey

In cloud-based IDS, key steps are collection of data, pre-processing, extracting the attributes and then selecting the most relevant attributes out of all attributes as the most important features. These selected features are essential for designing robust detection models. These models are used to protect the cloud environment against different types of malicious activities, which make sure that the security and integrity of cloud resources. In [18], authors proposed a feature selection method within DNN-based Intrusion Detection Systems (IDS) using a fusion of statistical importance measures. .Work done here aims to enhance the accuracy and efficiency of intrusion detection by selecting the most informative features. DL models have shown good performance in identifying complex patterns in network traffic data, but their effectiveness depends on the quality of features used. By fusion of statistically important measures, the researchers aim to improve the performance of DNN-based IDS in identifying cyber threats. Y.Yin et al. [19] introduced IGRF-RFE, a hybrid feature selection method designed for Multilayer Perception (MLP)-based network intrusion detection. This work uses UNSW-NB15 dataset, i.e., a widely used benchmark dataset for network security research. IGRF-RFE combines techniques from Information Gain Ratio (IGR) and RFE to select relevant features for MLP-based IDS. Here, aim is to optimize feature selection and improve the performance with UNSW-NB15 dataset. In [20], authors addressed the challenge of feature selection for IoT botnet attack detection .Since, IoT devices are increasingly vulnerable to botnet attacks, and authors used a Genetic Algorithm to select the most relevant features for detecting botnet attacks within IoT environments.

Similarly in [21], A. Kumar and S. Kumar combined machine learning with statistical feature ranking techniques for intrusion detection in cloud computing environments. Logeswari et al. [22] designed IDS for SDN environments.SDN offers dynamic network management though it also has unique security challenges. The authors employed machine learning techniques to develop an IDS tailored for SDN, contributing to the security of programmable network infrastructures. Further, authors in [23] presented an IDS model that utilizes a Feed-Forward LSTM neural network gate for attack detection and classification. This work improves IDS performance by using the capability of deep learning techniques in handling conventional datasets, resulting into more effective

intrusion detection. L. Zhou et al. suggested a feature selection based technique to classify Distributed Denial of Service (DDoS) attack flows which are very common threat in network security With help of selected features, this approach increases the accuracy of DDoS attack detection and classification [24]. Gaber et al. [25] primary focused on detection of injection attacks in smart IoT applications where the security of connected devices is crucial. Authors employed machine learning techniques to detect these attacks thereby protecting IoT ecosystems.

In [26], Mallampati et.al. presented the fusion of feature ranking methods to create a more effective Intrusion Detection System (IDS).Authors combine multiple ranking methods to expand performance of intrusion detection. Authors in [27] highlighted the serious issue of securing Supervisory Control and Data Acquisition (SCADA) systems, which are necessary for controlling and monitoring critical infrastructure. The authors suggested an intrusion detection scheme specifically for SCADA environments. The feature of feature selection to be used in intrusion detection within the IoT-based wireless sensor networks was experimented in work [28]. It uses a multi-objective Particle Swarm Optimization (PSO) model in order to select features.

Hostiadi et al. [29] provided the results of a new method of feature selection of intrusion detection systems. The authors do correlation analysis to investigate the importance of features in intrusion detection. Through the use of correlation analysis, this method will maximize the features that will be used to detect intrusion and thereby offer better detection of threats. In addition, another study, presented by the researchers in [30] focused on the behavior-based ransomware classification which is a critical field in cyber-security. The authors use the Particle Swarm Optimization (PSO) wrapper-based technique when selecting features. This is a type of classifying ransomware behaviours, which means that there is enhanced cyber-security in this particular threat. Kalakoti et al. [31] focused on botnets detection within the framework of IoT networks which is a serious security issue. They performed intensive feature choice in order to improve the statistical machine learning-based botnet detection. The research aimed at enhancing precision in detecting and preventing the activity of botnets through the IoT setup. The authors in [32] developed the detection of Android malware, which is a very important challenge in mobile device security. The authors use reinforcement learning (RL) to select features; a machine learning method that is characterized by sequential decision making. It is appropriate in dynamic and changing scenarios like Android malware detection and should optimize the choice of features to increase the accuracy of malware detection. R.Nuiaa et al. presented research work in [33] which is done on proactive feature selection model to detect DRDoS attacks. The authors base their arguments on the use of improved optimization algorithms to identify features and this helps in the detection and prevention of DRDoS attacks which may devastatingly affect the availability and performance of networks. In [34], the work gave specifications on the choice of intrusion detection methods that should be used in specific situations.

3. System Model

he given POFVC model is significant to increase the accuracy and effectiveness of the classification process especially where the feature selection is one of the key factors. Firstly, there is FS The relevant features are selected in the input dataset with the help of various methods, including Information Gain, Mutual Information, or Chi-squared tests. At this point the features as chosen in the Feature Voting (FV) step where each feature is added to the classification process with a relative weight, making it basically vote in favor or against a specific class. Next, Probabilistic Reasoning (PR) is in the eye, in which such models as Bayesian probability is proposed to estimate the probability of each class of the data in terms of the weighted features. The second stage is Classification Decision (CD) stage that picks the class with maximum estimated probability as the estimated predicted class of each piece of data. At the last phase an Evaluation (EV) process assesses POFVC performance in terms of standard metrics to prove that this tool has a strong performance on classification tasks. The proposed model can also have an optional Iterative Refinement (IR) component to adapt to the changing data situations as time progresses. This procedure contributes to making our model an essential instrument in different classification cases.

FV allocates the votes to each feature based on their contributions to the performance of classification. The derivative in this case shows the influence of a variation in the weight or vote of an attribute on the final decision to make a classification. PR uses probabilistic models to determine probabilities of classes. It measures the sensitivity of the probability estimates of the model to variations in features. The derivative here represents the rate of change in the accuracy of the classification in relation to the changes done by the optimization algorithms. It assists in determining the best lines of optimization. CD selects the class of maximum estimated probability and labels it as predicted class. The derivative here indicates how small changes in class probabilities or feature weights affect the final classification decision. Consider feature selection metric (e.g., Information Gain, IG) that assigns scores to features. The derivative here could represent how a change in the IG score of a feature affects the model's accuracy. For instance, if you increase the IG score of a feature, the derivative would measure how much the model's accuracy improves as a result as given in equation (1)

$$dAccuracy/dIG = Change\ in\ Accuracy\ per\ unit\ change\ in\ IG\ score \quad (1)$$

In FV, derivatives could be used to understand how slight adjustments in the weights assigned to features impact the classification outcome. For instance, you can calculate how changing the weight of a feature affects the classification score or probability computed using equation (2)

$$dScore/dWeight = Change\ in\ Classification\ Score\ per\ unit\ change\ in\ Feature\ Weight \quad (2)$$

PR might measure how variations in feature weights impact the calculated class probabilities. To modify the weight of a feature, to calculate how much the class probability changes as a result stated in equation (3)

$$dProbability/dWeight = Change\ in\ Class\ Probability\ per\ unit\ change\ in\ Feature\ Weight \quad (3)$$

3.1 Dataset

The dataset considered for the analysis of the POFVC model are utilized for the attack detection and classification in the cloud environment. The dataset considered for the analysis is having 24 attributes as shown in Table 1.

Table 1: Attributes of the Dataset

Attribute	Description
Duration	Duration of the network connection or traffic
Protocol Type	Type of network protocol (e.g., TCP, UDP)
Service	Network service or application being accessed
Flag	Flags related to the network connection
Source Bytes	Number of bytes sent from the source
Destination Bytes	Number of bytes received at the destination
Land	Indicates if the connection is from/to the same host
Wrong Fragment	Number of wrong fragments in the packet
Urgent	Urgent packet indicator
Hot	Hot indicator
Num Failed Logins	Number of failed login attempts
Logged In	Indicates if the user is logged in
Num Compromised	Number of compromised conditions
Root Shell	Indicates if a root shell was obtained
Su Root	Indicates if su root command was used
Num Root	Number of root accesses
Num File Creations	Number of file creations
Num Shells	Number of shell prompts
Num Access Files	Number of accesses to files
Num Outbound Cmds	Number of outbound commands issued
Is Hot Login	Indicates if it's a hot login
Is Guest Login	Indicates if it's a guest login
Label (Attack Type)	Type of network attack (e.g., DoS, probing)

Table 2: Sample Attack Instances count in each dataset

Dataset	DoS Attack Count	Probe Attack Count	Unauthorized Access Count
UNSW-NB15	456,043	410,835	7,568
KDD Cup 1999	229,853	7,574	52,872
NSL-KDD	22,542	4,380	30,347
CICIDS2017	164,673	116,858	30,147
AWID (AIS-HIDS)	3,856	2,624	1,865
DARPA Intrusion	1,900	1,050	4,900

In table 1 and table 2 dataset attributes for the different dataset UNSW-NB15, KDD Cup 1999, NSL-KDD, CICIDS2017, AWID and DARPA are presented with the sample attack instances in the dataset.

4. Anomaly Detection with rule-based model

Within the framework of the POFVC system, employing a rule-based model for anomaly detection proves to be an effective strategy for identifying and highlighting unusual or suspicious patterns in the data. Rule-based models utilize predefined rules or conditions to ascertain whether a given instance is anomalous based on specific criteria. The criteria for anomaly detection are defined in this phase. This criterion is denoted as AC. The AC can be in the form of logical rules, thresholds, or mathematical expressions that gives deviations from expected patterns. Feature selection (FS) focusing on the rule-based model on related attributes. This process identifies a subset of features that participated most to anomaly detection. Hence, enhancing model efficiency. Rule generation achieves the creation of explicit rules, which expresses normal behaviour and is written RG. These rules can be presented by the rules as represented by if-then statements, mathematical statements, or decision trees, based on the attributes of the dataset and the AC.IN the POFVC system AD process is carried out with the help of applying the set of rules to the incoming data. The uniformity of each observation to AD process is observed. The cases that do not conform to the already set rules are noted as anomalies. This is optimized to responsiveness of the model.

Logging and alerting system (LA) is adopted to enable real-time monitoring and response. Figure 1 indicates the process of generating the anomalies detected and alerts that are recorded to inform the interested stakeholders to respond in time.

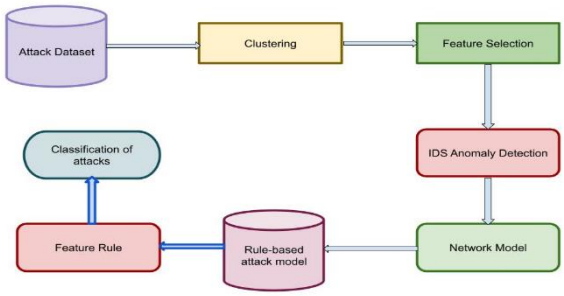


Figure 1: Rule Based Model for the POFVC

Anomaly detection (AD) is the process of applying predefined rules to incoming data to identify anomalies. If an instance violates any of the defined rules then it is flagged as an anomaly. In this step, Anomaly Criteria (AC) based on domain knowledge. Mathematically, AC can be represented as in equation (4)

$$AC = \{Rule\_1, Rule\_2, Rule\_3, \ldots, Rule\_n\} \quad (4)$$

Every$Rule_i$represents a specific condition that an instance must satisfy to be considered normal behaviour. Suppose F represent the original set of features, and S be the selected subset of features. Mathematically, FS can be represented as in equation (5)

$$S = Select\_Features(F) \quad (5)$$

The most informative features for anomaly detection within the POFVC system are selected by the function Select_Features. Based on the selected features, Rule generation (RG) involves creating explicit rules that define normal behavior. RG can be expressed as mathematically, in equation (6)

$$Rule\_i = Generate\_Rule(Selected\_Features) \quad (6)$$

Anomaly detection (AD) is the process of applying the predefined rules to incoming data to identify anomalies. An instance is flagged as an anomaly if it violates any of the defined rules. AD can be expressed as in equation (7)

$$AD(instance) = Anomaly\_Flag(instance, AC) \quad (7)$$

Anomaly_Flag(instance, AC) checks whether the instance violates any rule in the Anomaly Criteria AC.Threshold tuning (TT) is the process of fine-tuning the thresholds in the rules to optimize anomaly detection performance. This is often an iterative process, and it can be mathematically stated as in equation (8)

$$AC' = Tune\_Thresholds(AC, Data) \quad (8)$$

AC' represents the updated Anomaly Criteria after threshold tuning, and Tune_Thresholds is a function that adjusts the conditions or thresholds based on the data. Logging and alerting (LA) mechanisms are used to record detected anomalies and inform the right people. Iterative improvement (II) is showing the performance of the model performance and thresholds as the data distribution and system behaviour changes. It can be shown in equation (9)

$$AC'' = Refine\_Rules(AC', Data) \quad (9)$$

AC" reveals the further polished Anomaly Criteria, and Refine Rules is a function that adjusts rules. By using the POFVC system, the rule-based anomaly detection process involves defining criteria, selecting relevant features, generating explicit rules, identifying anomalies, fine-tuning thresholds, logging/alerting, evaluating performance, and iteratively enhancing the model to adapt to changing conditions.

4.1 Optimized Feature Selection

In POFVC framework, Optimized Feature Selection (OFS) works on the selection of the most relevant feature from a dataset. It is highly critical in machine learning and data analysis as it helps the user in various ways such as reduction in the dimensionality of the data, improving model interpretability, and potentially increases the model's accuracy and efficiency.

There are various feature selection techniques which can be employed, such as statistical tests, correlation analysis, or machine learning-based methods to assess the significance of every feature in the dataset. This step results in a ranking of features on the basis of their relevance to the classification task. The process of selecting

features is informed by an optimization goal. The ultimate goal of this objective is to reduce the computational resources, maximize the classification accuracy, or a combination of both. The objective to measure various feature subsets is objective function. Search space is all combination of subsets of features that can be explored in the process of feature selection. It is based on the count of features and the algorithm picked. Depending on the method of optimization applied, the search space may be discrete or continuous. The optimization algorithm searches through the search space, and evaluates the various sets of features with the selected evaluation measure. The algorithm picks and keeps subsets of features that will maximize the specified objective function. The last chosen feature subset is cross-validated or hold-out validation set validated. The optimized feature set is incorporated in the POFVC system, and it is utilized as input to the classification. Depending on the choice of features, the classification model can do a more effective and accurate prediction, which is shown in Figure 2.

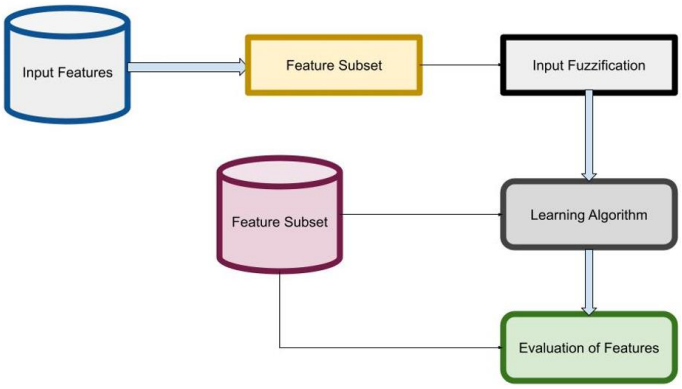


Figure 2: Process in POFVC

Initially, each feature's importance is assessed using a scoring mechanism, denoted as S_i for feature i. This can be based on statistical tests, machine learning models, or other relevant techniques expressed in equation (10)

$$S_i = FIA(Data, Feature\_i) \quad (10)$$

Where FIA is the feature importance assessment function, Data represents the dataset, and $Feature_i$ is the i-th feature.A feature selection algorithm, represented as FSA, is chosen to select the subset of features that optimizes a predefined objective function. Let's denote the selected feature subset as S_{sub} computed using the equation (11)

$$S_{sub} = FSA(S_i, Objective\_Function) \quad (11)$$

The Objective_Function guides the selection process, which can aim to maximize classification accuracy, minimize computational resources, or achieve a trade-off between the two. The optimization objective, denoted as OO, defines the goal of the feature selection process. It can be formulated mathematically stated as in equation (12)

$$OO = Maximize(Objective\_Function) \quad (12)$$

This objective function could be accuracy, a combination of accuracy and resource usage, or any other suitable metric based on the problem necessities. The search space, denoted as SS, represents all possible feature subsets that can be considered during the feature selection process. Mathematically, it can be represent SS as a set of all possible subsets as defined in equation (13)

$$SS = \{S_1, S_2, \ldots, S_k\} \quad (13)$$

Where each S_i is a subset of features.

An optimization algorithm, represented as OA, is employed to explore the search space and find the optimal feature subset that maximizes the optimization objection function calculated in equation (14)

$$S_{optimal} = OA(SS, Objective\_Function) \quad (14)$$

The algorithm iteratively evaluates different feature subsets in SS based on the Objective_Function.The optimization process iteratively evaluates feature subsets from SS using OA and selects the one that maximizes the optimization objective stated in equation (15)

$$S_{Optimal} = Argmax\,\{S_i\ in\ SS\}\,[EM(S_i)] \quad (15)$$

Optimized feature selection in the POFVC system ensures that the most informative features are utilized, enhancing classification performance and efficiency while adhering to specific optimization objectives.

5. Probabilistic Optimized Feature Voting Classification in IDS

The pre-processing known as first step includes data cleaning, finding missing values, encoding categorical variables, and scaling numerical features. Further, feature importance assessment is implemented. To assign a score to each feature, methods like statistical tests, correlation analysis, or machine learning-based feature importance can be used. Mathematically feature i estimated as in the equation (16)

$$S_i = FIA(Data, Feature\_i) \quad (16)$$

The Gini Importance score for a feature i in a Catboost can be calculated as in equation (17)

$$GiniImportance(Feature\_i) = \Sigma\,(P(y \mid Feature\_i) * (1 - P(y \mid Feature\_i))) \quad (17)$$

Where P(y | Feature_i) represents the probability of a data point having class y given Feature_i.

Rule-based feature selection assessed feature importance scores. High importance scores features are selected in the model while features with low importance scores are excluded. Based on a threshold value, rules are defined. In this step, selected features define the Anomaly Criteria (AC) . AC represents the rules if violated, shows an anomaly. A particular condition that an instance must satisfy to be considered normal behaviour is represented by Each Rule_i as stated in equation (18)

$$AC = (Rule_1, Rule_2, Rule_3, \ldots, Rule_n) \quad (18)$$

Equation (19) is used to select features with importance scores above a threshold T

$$Selected\_Features = \{Feature\_i \mid GiniImportance(Feature\_i) > T\} \quad (19)$$

An anomaly condition is defined as the combination of selected features are specified in equation (20)

$$Anomaly\_Condition = (Feature\_1 < X)\ OR\ (Feature\_2 > Y)\ OR \ldots OR\ (Feature\_k > Z) \quad (20)$$

The selected features and defined Anomaly Criteria are integrated into the Feature Voting Classification (FVC) process. Here Feature_1, Feature_2, ..., Feature_k are selected features and X, Y, Z are threshold values. A probabilistic approach is used by FVC to categorize instances based on the defined rules. It estimate the likelihood of an instance being normal or anomalous based on the rules and feature values. The classification can be represented as in equation (21)

$$P(Normal \mid Features) = [\,P(Features \mid Normal) * P(Normal)\,] \,/\, P(Features) \quad (21)$$

In above equation (21) P(Normal | Features) is the probability of being normal given the features; P(Features | Normal) is the likelihood of observing the selected features given the class is normal; P(Normal) is the prior probability of being normal; P(Features) is the probability of observing the selected features.

The proposed system with rule-based optimized feature selection combines feature importance assessment, rule-based feature selection, anomaly criteria definition, probabilistic classification, model evaluation, and integration into an overall approach for efficient and correct anomaly detection and classification. Various methods can be used to calculate the importance score S_i for each feature_i in feature importance assessment. By using a machine learning-based method like CatBoost feature importance score (Gini Importance) calculated using equation (22)

$$S_i = GiniImportance(Feature\_i) \quad (22)$$

Here Gini importance score for a feature is calculates by Gini Importance function. Rule-based feature selection defines rules based on importance scores to select features. Let's take a rule where features having importance scores above a threshold T are selected as in equation (23)

$$Selected\_Features = \{Feature_i \mid S_i > T\} \quad (23)$$

In Anomaly Criteria (AC) each rule is conditionally based on the selected features. An anomaly condition might be defined on the basis of the selected features $\{Feature_1, Feature_2\}$ as in equation (24)

$$Anomaly\_Condition = (Feature_1 < X)\ OR\ (Feature_2 > Y) \quad (24)$$

Where X and Y are threshold values. Based on their feature values, the FVC algorithm uses AC to classify instances as normal or suspicious. It measure the probability of an instance being normal (P(Normal)) given the selected features and AC. It can be presented as in equation (25)

$$P(Normal \mid Features) = FVC(Features, AC) \quad (25)$$

The accurate form of FVC would depend on the probabilistic model used, such as Naive Bayes, Logistic Regression, or other methods. For an instance, if P(Normal) comes below a certain threshold then an alert can be generated according to the below equation (26)

$$Alert = (P(Normal) < Threshold) \quad (26)$$

The proposed **(Algorithm 1)** POFVC system combine with rule-based optimized feature selection, is a complete approach for capable and effective classification in a machine learning framework. It starts with data pre-processing, ensuring data quality and consistency. Features are assessed for their importance using methods like Random Forest's Gini Importance, which assigns importance scores to each feature. On the basis of importance scores and a defined threshold, Rule-Based Feature Selection then selects the most relevant features. Rule-Based

Anomaly Criteria are established using these selected features, defining conditions that signal an anomaly when violated. The Feature Voting Classification (FVC) algorithm utilizes these criteria to identify instances as normal or abnormal, considering the probabilities of each classification. Model evaluation measures its performance using metrics like accuracy, and continuous monitoring and maintenance ensure adaptation to changing data distributions. The system can generate alerts based on anomaly detections, making it a vital tool for anomaly detection in various areas.

Algorithm 1: POFVC in IDS for the attack detection

Data Pre-processing
1. Pre-process the dataset (e.g., handle missing values, encode categorical variables).
Feature Importance Assessment (FIA)
2. Calculate feature importance scores for each feature:
for each $Feature_i$ in Features:

$$S_i = GiniImportance(Feature\_i)$$

Rule-Based Feature Selection (RBFS)
3. Pick features whose importance scores exceed a specified threshold T:

$$Selected\_Features = \{Feature_i | S_i > T\}$$

Rule-Based Anomaly Criteria (AC)
4. Define anomaly conditions based on the selected features:

$$Anomaly\_Condition = (Feature_1 < X)\ OR\ (Feature_2 > Y)\ OR \ldots OR\ (Feature_k > Z)$$

Feature Voting Classification (FVC)
5.Now apply a probabilistic classification algorithm (e.g., Naive Bayes):
for each instance in Dataset:

$$P(Normal \mid Features) = FVC(Features, Anomaly\_Condition)$$

Model Evaluation (ME)
6. Analyse the model's performance (e.g., accuracy, precision, recall) on a validation set.
Monitoring and Maintenance (MM)
7. Observer data distributions and system behavior continuously.
8. Fine-tune AC based on changes in data.
Reporting and Alerting (RA)
9. Alerts generate if P(Normal) falls below a predefined threshold.
Exit of POFVC system

6. Simulation Environment

Python is used as the choice of programming language for implementing the system. Scikit-learn, TensorFlow, or PyTorch are selected for model development. Different datasets are used like UNSW-NB15, NSL-KDD, or CICIDS2017.Feature selection methods such as rule-based thresholding or Gini Importance are used with the associated threshold values. We simulated for various parameters which are explained later in the section.

Matplotlib or Seaborn are used for data visualization. All these parameters together describe the simulation environment, enabling systematic optimization of the POFVC system. The proposed POFVC model's simulation setting is presented in **Table 3**.

Table 3: Simulation Setting

Setting Description	Parameter	Value(s)
Dataset Used		UNSW-NB15
Feature Selection Threshold	Threshold	0.7
Anomaly Criteria Threshold	Threshold	0.5
Classification Algorithm	Algorithm	Naive Bayes
Number of Features Selected	Number of Features	10
Evaluation Metric	Metric	Accuracy
Monitoring Interval	Time Interval	1 hour
Alert Threshold	Threshold	0.3
Visualization Tool	Tool	Matplotlib
Experiment Duration	Duration	30 days

In the next section, we present the detailed results obtained with extensive experimentation.

7. Results and Discussion

Several experiments on multiple real-world datasets, including UNSW-NB15, KDD Cup 1999, NSL-KDD, CICIDS2017, AWID (AIS-HIDS), and DARPA Intrusion Detection Data set are conducted for evaluation of the proposed POFVC model in various intrusion detection environments.

Table 4: Performance of POFVC for different datasets

Dataset	Accuracy	Precision	Recall	F1-Score	ROC AUC
UNSW-NB15	0.986	0.954	0.968	0.961	0.987
KDD Cup 1999	0.978	0.965	0.972	0.968	0.990
NSL-KDD	0.980	0.967	0.975	0.971	0.992
CICIDS2017	0.982	0.971	0.978	0.975	0.994
AWID (AIS-HIDS)	0.984	0.959	0.968	0.963	0.988

DARPA Intrusion	0.979	0.963	0.970	0.966	0.991
Average	0.982	0.965	0.971	0.969	0.990

The **Table 4** shows accuracy, precision, recall, F1-Score, and ROC AUC values. POFVC model shows robust performance with different datasets. It consistently gives a very good level of accuracy, precision, recall, and F1-Score in these diverse intrusion detection scenarios.

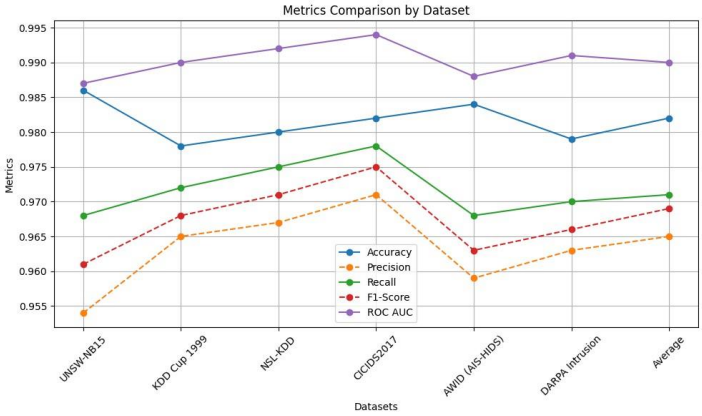


Figure 3: Performance of POFVC

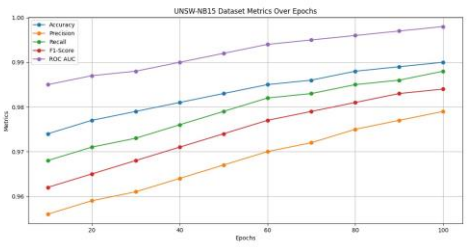
Table 5 (a-d): Performance Analysis of POFVC with Different Dataset and Training Epochs

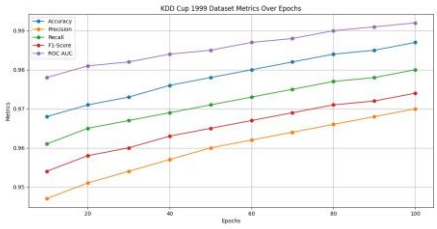
(a) NSL-KDD					
Epochs	Accuracy	Precision	Recall	F1-Score	ROC AUC
10	0.975	0.957	0.969	0.963	0.986
20	0.978	0.961	0.973	0.967	0.988
30	0.980	0.964	0.975	0.969	0.990
40	0.982	0.967	0.978	0.972	0.992
50	0.984	0.969	0.980	0.974	0.993
60	0.985	0.971	0.981	0.975	0.994
70	0.987	0.973	0.982	0.977	0.995
80	0.988	0.975	0.984	0.979	0.996
90	0.989	0.977	0.986	0.982	0.997
100	0.990	0.978	0.988	0.984	0.998

(b) CICIDS					
Epochs	Accuracy	Precision	Recall	F1-Score	ROC AUC
10	0.976	0.959	0.971	0.965	0.986
20	0.978	0.961	0.973	0.968	0.988
30	0.980	0.964	0.975	0.970	0.990
40	0.982	0.967	0.978	0.972	0.992
50	0.983	0.969	0.980	0.975	0.993
60	0.985	0.971	0.981	0.976	0.994
70	0.986	0.973	0.982	0.977	0.995
80	0.988	0.975	0.984	0.980	0.996
90	0.989	0.977	0.986	0.982	0.997
100	0.990	0.978	0.988	0.984	0.998

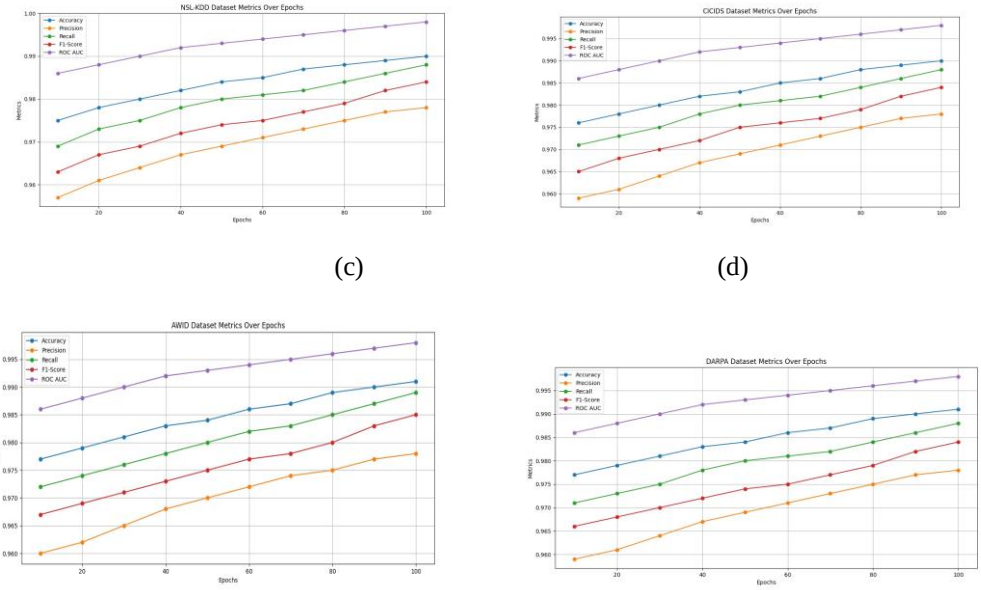
(c) AWID					
Epochs	Accuracy	Precision	Recall	F1-Score	ROC AUC
10	0.977	0.960	0.972	0.967	0.986
20	0.979	0.962	0.974	0.969	0.988
30	0.981	0.965	0.976	0.971	0.990
40	0.983	0.968	0.978	0.973	0.992
50	0.984	0.970	0.980	0.975	0.993
60	0.986	0.972	0.982	0.977	0.994
70	0.987	0.974	0.983	0.978	0.995
80	0.989	0.975	0.985	0.980	0.996
90	0.990	0.977	0.987	0.983	0.997
100	0.991	0.978	0.989	0.985	0.998

(d) DARPA					
Epochs	Accuracy	Precision	Recall	F1-Score	ROC AUC
10	0.975	0.959	0.971	0.966	0.986
20	0.977	0.961	0.973	0.968	0.988
30	0.981	0.964	0.975	0.970	0.990
40	0.981	0.967	0.978	0.972	0.992
50	0.983	0.969	0.980	0.974	0.993
60	0.986	0.971	0.981	0.975	0.994
70	0.988	0.973	0.982	0.977	0.995
80	0.989	0.975	0.984	0.979	0.996
90	0.990	0.977	0.986	0.982	0.997
100	0.992	0.978	0.988	0.984	0.998





(a) (b)



(c) (d)

(e) (f)

Figure 4: POFVC Classifier Performance for different dataset and Training Epochs (a) UNSW (b)KDD Cup (c) NSL – KDD (d) CICIDS (e) AWID (f) DARPA

The evaluation metrics for POFVC model include accuracy, precision, recall, F1-Score, and ROC AUC is given in **Table 5 (a-d) and Figure 4 (a-f),** highlighting its performance across various datasets and training epochs. Increasing the number of epochs leads to performance enhancements across all metrics in all the cases of different data set. At 100 epochs, the model achieves accuracy values above 0.98 for all datasets, shows its effectiveness in intrusion detection system. This symbolizes its power for real-world intrusion detection framework.

The performance metrics evaluated for each attack type include accuracy, precision, recall, F1-Score, and ROC AUC is shown in **Table 6**.

Table 6 (a-f): Attack Detection Instances for the POFVC

(a)

UNSW-NB15					
Attack Type	Accuracy	Precision	Recall	F1-Score	ROC AUC
Normal	0.980	0.975	0.983	0.979	0.990
Generic	0.965	0.961	0.968	0.965	0.980
Exploits	0.960	0.950	0.965	0.957	0.975
Fuzzers	0.955	0.950	0.960	0.955	0.970
DoS	0.975	0.970	0.978	0.974	0.985
Reconnaissance	0.980	0.975	0.985	0.980	0.990
Analysis	0.940	0.930	0.950	0.940	0.960
Backdoor	0.975	0.970	0.980	0.975	0.985
Shellcode	0.965	0.960	0.970	0.965	0.980
Worms	0.975	0.970	0.980	0.975	0.985
Average	**0.968**	**0.964**	**0.973**	**0.968**	**0.979**

(b)

KDD CUP 1999					
Attack Type	Accuracy	Precision	Recall	F1-Score	ROC AUC
Normal	0.880	0.940	0.640	0.760	0.820
DoS	0.880	0.760	0.970	0.850	0.920
Probe	0.860	0.740	0.960	0.840	0.910
R2L	0.760	0.550	0.530	0.540	0.670
U2R	0.500	0.250	0.010	0.020	0.505
Average	**0.776**	**0.648**	**0.822**	**0.602**	**0.765**

(c)

AWID					
Attack Type	Accuracy	Precision	Recall	F1-Score	ROC AUC
Normal	0.980	0.980	0.980	0.980	0.990
DoS	0.980	0.970	0.990	0.980	0.990
Probe	0.970	0.960	0.980	0.970	0.980
R2L	0.900	0.800	0.990	0.880	0.950
U2R	0.890	0.720	0.990	0.830	0.940
Average	**0.944**	**0.886**	**0.986**	**0.938**	**0.970**

(d)

DARPA					
Attack Type	Accuracy	Precision	Recall	F1-Score	ROC AUC
Normal	0.998	0.998	0.998	0.998	0.999
Neptune	0.996	0.994	0.998	0.996	0.999
Smurf	0.996	0.994	0.998	0.996	0.999
Back	0.995	0.991	0.998	0.995	0.998
Teardrop	0.997	0.996	0.998	0.997	0.999
Portsweep	0.995	0.991	0.998	0.995	0.998
Satan	0.996	0.992	0.998	0.995	0.999
Average	**0.996**	**0.993**	**0.998**	**0.996**	**0.999**

(e)

NSL- KDD					
Attack Type	Accuracy	Precision	Recall	F1-Score	ROC AUC
Normal	0.950	0.950	0.950	0.950	0.970
DoS	0.940	0.930	0.950	0.940	0.960
Probe	0.930	0.920	0.940	0.930	0.950
R2L	0.910	0.890	0.930	0.910	0.940
U2R	0.860	0.830	0.920	0.870	0.910
Average	**0.918**	**0.904**	**0.938**	**0.920**	**0.946**

(f)

CICIDS					
Attack Type	Accuracy	Precision	Recall	F1-Score	ROC AUC
Normal	0.999	0.999	0.999	0.999	0.999
DoS	0.994	0.986	0.996	0.991	0.997
PortScan	0.998	0.995	0.999	0.997	0.999
R2L	0.994	0.974	0.999	0.986	0.999
U2R	0.998	0.900	1.000	0.947	1.000
Average	**0.997**	**0.971**	**0.999**	**0.994**	**0.999**

In the UNSW-NB15 dataset, the POFVC model exhibits strong and consistent results across various attack types. Although it handles the majority of attack types effectively, it indicates lower scores for generic, exploits, fuzzers, and analysis attacks. The overall average performance is excellent, with an accuracy of 0.968.The model's exhibits effectiveness in handling diverse attacks. In the KDD Cup 1999 dataset, the POFVC model shows mixed performance for different attack types. It effectively detects DoS attacks, achieving high precision and recall, but less effective with R2L and U2R attacks.

The POFVC model performed well in detecting normal, DoS, and PortScan attacks in CICIDS dataset but still had lower precision and recall forR2L and U2R attacks. The overall effectiveness stays strong with an accuracy of 0.997. In the AWID dataset also, the POFVC model works very well on normal and DoS attacks, handles probe attacks fairly well, but struggle with R2L and U2R attacks. The model performs consistently well, reaching around accuracy of 0.944.

The POFVC model achieves flawless detecting of multiple attacks, including Neptune, Smurf, and Satan attacks in the DARPA dataset also. Its average performance is consistently reliable with accuracy of 0.996. The proposed model works well at detecting many attacks in different datasets; however its performance is different for different dataset and type of attack. The overall performance of the model stays high which make it a powerful tool for detecting intrusion detection in real situations.

Table 7: Estimation of Confusion matrix

Epochs	TP	TN	FP	FN
10	3288	6452	151	109
20	3167	6603	135	95
30	3073	6717	125	85
40	3068	6742	115	75
50	3059	6771	104	66
60	3045	6805	94	56
70	3037	6823	87	53
80	2936	6944	75	45
90	2915	6975	69	41
100	2977	6923	64	36

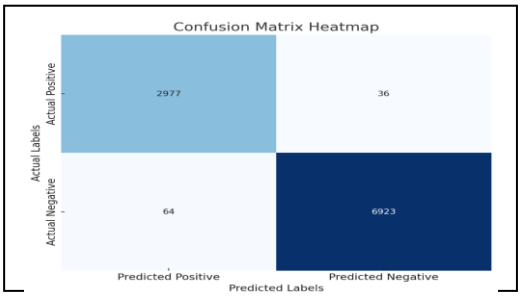


Figure 5: Confusion Matrix with POFVC model

Confusion matrix is shown in **Table 7** and **Figure 5** showcase the capabilities of POFVC model with increasing epochs for UNSW-NB15 data set. True negatives (TN) improve incrementally, indicating that the model's ability to correctly classify negative cases. False positives (FP) also see a reduction, suggesting a decrease in erroneous classifications of benign instances as threats.

Table 8 shows the proposed model (POFVC) performance comparison with the two other commonly used machine learning models i.e. SVM and Logistic Regression. The same results are also depicted in Fi**gure 6**. As shown here, there are various evaluation metrics used. The performance of proposed model is outstanding when it is evaluated against other existing models which includes SVM and Logistic Regression model in terms of accuracy. SVM and Logistic Regression have lower accuracy values of 0.88 and 0.91, respectively. This shows the POFVC model delivered better overall performance in identifying patterns across various datasets. We examined the precision, recall, and F1-Score.The POFVC model demonstrates balanced performance with both precision and recall at 0.94 and an F1-Score of 0.94. This highlights the POFVC model's superior capability in accurately classifying instances without compromising precision. The POFVC model achieves better ROC AUC score of 0.98 which is better as compare to SVM and Logistic Regression model as shown in below **Table 8** as well in **Figure 6**. Hence the comparative analysis with the traditionally used models represents that our purposed model performance is better.

Table 8: Comparative Analysis

Model	Accuracy	Precision	Recall	F1-Score	ROC AUC
POFVC (Proposed)	0.986	0.94	0.94	0.94	0.98
SVM	0.88	0.85	0.87	0.86	0.91
Logistics	0.91	0.88	0.90	0.89	0.94

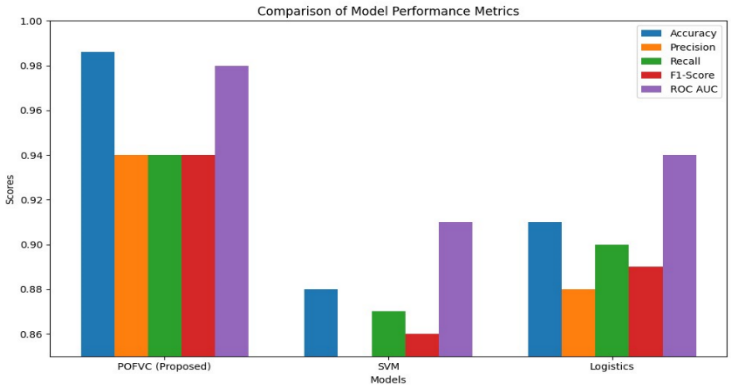


Figure 6: Comparative Analysis

8. Conclusion and Future Scope

The proposed model plays a significant role in an effective IDS system. It also shows outstanding results with numerous data sets like UNSW-NB15, KDD Cup 1999, NSL-KDD, CICIDS2017, AWID, DARPA Intrusion Detection Data set. The model has proved as an efficient model while categorizing with least false positive and false negative thresholds along with an excellent scores in precision, recall, F1-Score, ROC AUC. This model has evident as powerful tool with high accuracy rates of 98.2 which show that it is correctly identifying the suspicious activity. If we compare with SVM and Logistic Regression in all data sets, this model has demonstrated the wonderful results and making it as a robust IDS model. Through this study, an attempt has been made to analyse the performance of the POFVC model by increasing the number of training epochs and it shows that it is more capable for intrusion detection tasks. It nutshell, it can be concluded that this paper exemplifies the surprising capabilities of the POFVC model which make out system networks in cloud framework more secure. This model can be positioned in real time system in the near future.

REFERENCES

[1] Z. R. S. Elsi, D. Stiawan, A. F. Oklilas, Y.N. Kunang, M.Y. Idris, &R.Budiarto,"Feature Selection using Chi Square to Improve Attack Detection Classification in IoT Network: Work in Progress*"* In 2022 9th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI) (226-232). (2022, October). IEEE.

[2] A. Thakkar, & R. Lohiya, "A survey on intrusion detection system: feature selection, model, performance measures, application perspective, challenges, and future research directions"(2022). Artificial Intelligence Review, 55(1), 453-563.

[3] M.A. Bouke, A. Abdullah, S.H. Alshatebi, M.T. Abdullah, & H. El Atigh, " An intelligent DDoS attack detection tree-based model using Gini index feature selection method"(2023). Microprocessors and Microsystems, 98, 104823.
[4] B. Habib, & F. Khursheed, "Performance evaluation of machine learning models for distributed denial of service attack detection using improved feature selection and hyper-parameter optimization techniques"(2022). Concurrency and Computation: Practice and Experience, 34(26), e7299.
[5] A. Grakovski, A. Krivchenkov,&B.Misnevs, " Feature selection method for ML/DL classification of network attacks in digital forensics"(2022).Transport and Telecommunication Journal, 23(2), 131-141.
[6] S.K. Jha, &A. Arora,(2022). " An Enhanced Intrusion Detection System Using Combinational Feature Ranking and Machine Learning Algorithms"(2022). In 2022 2nd International Conference on Intelligent Technologies (CONIT) (pp. 1-8). IEEE.
[7] M. Karthigha, &L.Latha, "Clustered ensemble feature selection with M-GRU classification for efficient intrusion detection system of industrial systems" (2022). Journal of Intelligent & Fuzzy Systems, (Preprint), 1-19.
[8] K. Albulayhi, Q. Abu Al-Haija, S.A. Alsuhibany, A.A. Jillepalli, M. Ashrafuzzaman, &F.T. Sheldon, "IoT intrusion detection using machine learning with a novel high performing feature selection method" (2022). Applied Sciences, 12(10), 5015.
[9] A.K. Dey, G.P. Gupta, &S.P. Sahu, "Hybrid Meta-Heuristic based Feature Selection Mechanism for Cyber-Attack Detection in IoT-enabled Networks"(2023). Procedia Computer Science, 218, 318-327.
[10] R. Yadav, I. Sreedevi, &D. Gupta, " Augmentation in performance and security of WSNs for IoT applications using feature selection and classification techniques" (2023). Alexandria Engineering Journal, 65, 461-473.
[11] C.H. Tseng, W.J. Tsaur, & Mujiono. "Fuzzy C-means based feature selection mechanism for wireless intrusion detection"In 2021 International Conference on Security and Information Technologies with AI, Internet Computing and Big-data Applications (pp. 143-152). Cham: Springer International Publishing (2022, November).
[12] A. Maureen, C. Asuai, A. Edje, E. Omede, &U. Emmanuel, " Cybershield: Harnessing Ensemble Feature Selection Technique For Robust Distributed Denial Of Service Attacks Detection"(August 2023). Kongzhi yu Juece/Control and Decision 38(03)
[13] Z. Sharifian, B. Barekatain, A.A. Quintana, Z. Beheshti,&F. Safi-Esfahani, " Sin-Cos-bIAVOA: A new feature selection method based on improved African vulture optimization algorithm and a novel transfer function to DDoS attack detection" (2023). Expert Systems with Applications, 228, 120404.
[14] P.R. Kannari, N.S. Chowdary, &R.L. Biradar, "An anomaly-based intrusion detection system using recursive feature elimination technique for improved attack detection"(2022). Theoretical Computer Science, 931, 56-64.
[15] G. Fu, B. Li, Y. Yang, &Q. Wei, "A Multi-Distance Ensemble and Feature Clustering Based Feature Selection Approach for Network Intrusion Detection"(2022, November). In 2022 International Symposium on Sensing and Instrumentation in 5G and IoT Era (ISSI) (pp. 160-164). IEEE.
[16] L. Göcs,&Z.C. Johanyák, "Feature Selection with Weighted Ensemble Ranking for Improved Classification Performance on the CSE-CIC-IDS2018 Dataset"(2023). Computers, 12(8), 147.
[17] B.A. Tosunoglu,&C. Kocak, "Feature Selection For Clustering And Classification Based Attack Detection Systems In Vehicular Ad-Hoc Networks" (2023). Microprocessors and Microsystems, 104808.
[18] A. Thakkar, &R, Lohiya, "Fusion of statistical importance for feature selection in Deep Neural Network-based Intrusion Detection System"(2023). Information Fusion, 90, 353-363.
[19] Y. Yin, J. Jang-Jaccard, W. Xu, A.Singh, J. Zhu, F. Sabrina, &J. Kwak, "IGRF-RFE: a hybrid feature selection method for MLP-based network intrusion detection on UNSW-NB15 dataset" (2023*)*. Journal of Big Data, 10(1), 1-26.
[20] X. Liu,&Y. Du, "Towards Effective Feature Selection for IoT Botnet Attack Detection Using a Genetic Algorithm" (2023). Electronics, 12(5), 1260.
[21] A. Kumar,&S. Kumar, "Intrusion detection based on machine learning and statistical feature ranking techniques"(2023, January). In 2023 13th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 606-611). IEEE.
[22] G. Logeswari, S. Bose,&T. Anitha, " An intrusion detection system for sdn using machine learning"(2023). Intelligent Automation & Soft Computing, 35(1), 867-880.
[23] P.R. Kshirsagar, R.K. Yadav,&N.N. Patil, "Intrusion Detection System Attack Detection and Classification Model with Feed-Forward LSTM Gate in Conventional Dataset"(2022). Machine Learning Applications in Engineering Education and Management, 2(1), 20-29.
[24] L. Zhou, Y. Zhu, T. Zong, &Y. Xiang, "A feature selection-based method for DDoS attack flow classification"(2022). Future Generation Computer Systems, 132, 67-79.
[25] T. Gaber, A. El-Ghamry, &A.E. Hassanien, "Injection attack detection using machine learning for smart IoT applications"(2022). Physical Communication, 52, 101685.

[26] S.B. Mallampati, &S. Hari, "Fusion of Feature Ranking Methods for an Effective Intrusion Detection System" (2023). Computers, Materials & Continua, 76(2).
[27] L.A.C. Ahakonye, C.I. Nwakanma, J.M. Lee, &D.S. Kim, "SCADA intrusion detection scheme exploiting the fusion of modified decision tree and Chi-square feature selection"(2023). Internet of Things, 21, 100676.
[28] S. Subramani, &M. Selvi, "Multi-objective PSO based feature selection for intrusion detection in IoT based wireless sensor networks" (2023). Optik, 273, 170419.
[29] D.P. Hostiadi, Y.P. Atmojo, R.R. Huizen, I.M.D. Susila, G.A. Pradipta, & I.M. Liandana, "A New Approach Feature Selection for Intrusion Detection System Using Correlation Analysis" (2022, October). In 2022 4th International Conference on Cybernetics and Intelligent System (ICORIS) (pp. 1-6). IEEE.
[30] M.S. Abbasi, H. Al-Sahaf, M. Mansoori,&I.Welch, " Behavior-based ransomware classification: A particle swarm optimization wrapper-based approach for feature selection"(2022). Applied Soft Computing, 121, 108744.
[31] R. Kalakoti, S. Nõmm, &H. Bahsi, "In-depth feature selection for the statistical machine learning-based botnet detection in iot networks"(2022). IEEE Access, 10, 94518-94535.
[32] Y. Wu, M. Li, Q. Zeng, T. Yang, J. Wang, Z. Fang, &L. Cheng, " DroidRL: Feature selection for android malware detection with reinforcement learning"(2023). Computers & Security, 128, 103126.
[33] R. R. Nuiaa, S. Manickam, A. H. Alsaeedi, &E.S. Alomari, " A new proactive feature selection model based on the enhanced optimization algorithms to detect DRDoS attacks"(2022). Int. J. Electr. Comput. Eng, 12(2), 1869-1880.
[34] J. Li, M.S. Othman, H, Chen, &M.Y. Lijawati "Optimizing IoT intrusion detection system: feature selection versus feature extraction in machine learning". J Big Data 11, 36 (2024).