International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Building an Intrusion Detection System on Ecommerce Data using Regression Analysis

Praveen Kumar Shukla[1], Dr. C. S. Raghuvanshi[2], Dr. Hari Om Sharan[3]

Submitted: 14/09/2023 **Revised**: 23/10/2023 **Accepted**: 07/11/2023

Abstract: A system for anomaly-based intrusion detection learns to identify acceptable network behaviour in order to detect intrusion. When anomalous network behaviour is observed outside of its training sets, it then issues a warning. Administrators utilize the Network Intrusion Detection and Prevention System to identify network security vulnerabilities in their organizations by detecting and blocking a number of well-known network attacks. It is more crucial than ever to identify network anomalies and cyberattacks since they aid in the creation of an efficient intrusion detection system, which is necessary for contemporary security. The Canadian Institute of Cyber Security published a new data set called CICIDS2019 network data set, which fixed the NSL-KDD issue. The research's Network Intrusion Detection dataset can be downloaded for free from Kaggle. The dataset is standardised after being pre-processed to eliminate cells with null values. Based on the networking facts, a variety of computational techniques have been used to determine whether or not an intrusion has occurred, including classic ML and ensemble learning models. Classic machine learning methods like AdaBoost, Naive Bayes, K Nearest Neighbour, Support Vector Machine, and Logistic Regression are employed in this work. The Ad, K Nearest Neighbour, Naive Bayes, Support Vector Machine, and Logistic Regression models are all developed into the proposed model. According to the accuracy, precision, recall, and f-measure experimental findings from the NSL-KDD dataset used in this work, the proposed system outperforms the existing methods.

***Keywords*:** *Intrusion Detection System, Machine Learning, LR, K Nearest Neighbour, SVM, Network Security.*

1. Introduction

The Internet has become an indispensable part of our daily lives, and the prevalence of web applications is on the rise. Consequently, there has been an increase in emphasis on network security [1]. One of the most important aspects of network security research is recognising unusual network behaviour. Systems for detecting intrusions are used to examine network traffic and identify questionable activity. Typically, anomaly- and signature-based detection systems are used to classify IDSs [2]. Reliance on technology has become essential in today's world. The world has been revolutionised by technology, yet everything has pros and cons. Technology breakthroughs have led to a sharp rise in cybercrime, and criminal users are using highly skilled methods to carry out their illicit operations by breaking into networks and carrying out harmful tasks that are started by malware of any kind [3]. Consequently, network security is essential and needs to be protected from malevolent users [4]. For this reason, intrusion detection systems were created to find breaches in computer networks. Any unauthorised network or system action that aims to jeopardise the data's confidentiality, integrity, or availability is called an intrusion. Because of the recent increase in the frequency and intensity of network attacks, intrusion detection systems (IDS) are now an essential part of networks and organisations [5]. Intrusion detection systems are an essential part of the defensive tactics used to shield networks and computer systems from invasions. IDS is a technique for keeping an eye on, spotting, and assessing unusual behaviours in networked environments that are thought to be against security guidelines [6]. This system's primary objective is to detect a wider variety of malware than a traditional firewall. As a result of the exponential rise of computer networks and technology, attacks on networks are becoming more frequent, making security one of the most pressing issues facing contemporary society. Using intrusion detection systems and a number of other technological solutions that can identify and stop possible network attacks is one effective way to address this problem and improve network security. One potential technology that keeps an eye out for hostile activity or rule breaches on a network is the intrusion detection system (IDS) [7]. IDS can stop a malicious party in a smart grid from using network flaws to access nodes without authorization. IDS also stops the exploitation of grid resources. Intrusion detection systems can be categorised into three groups: anomaly-based, specification-based, and signature-based. It is feasible to detect hostile cyberattack behaviour patterns with signature-based intrusion detection systems. On the other hand, only malicious activity deviations can be detected by specification IDS. Statistical methods are utilised in anomaly-based

[1,2,3] *Department of Computer Science & Engineering, FET, Rama University, Kanpur 209217, INDIA*
Corresponding Author: [2] *drcsraghuvanshi@gmail.com*

intrusion detection systems to differentiate between harmful and good conduct [8].

Most research has concentrated on the benefits and drawbacks of anomaly, specification, and signature data-based intrusion detection systems (IDSs). The superior capacity of anomaly-based intrusion detection systems to detect zero-day or multi-stage attacks sets them apart from other types of intrusion detection systems. Furthermore, these technologies [9–11] enable real-time cyber threat identification in networks like smart grids. Aside from this, the capacity of anomaly-based IDS to identify multi-step, integrated, and complicated attacks makes them superior to other IDS. A number of concerns, such as anomaly-based IDS's poor detection rate and high rates of false alarms and missing detections, need to be addressed despite the technology's advantages. As a result, several methods have been put out to increase anomaly-based intrusion detection system's efficiency in recognising and categorising various networks, such as smart grids. One suggested remedy is to use artificial intelligence methods, including machine learning models [12]. However, the great majority of these investigations produced data with significant rates of misidentification and false alarms. Furthermore, these tactics' effectiveness has only been assessed using a restricted range of criteria, such accuracy. Furthermore, a number of these models lacked optimisation, and the pre-processed datasets used were improper [13]. Furthermore, part of this study examined a small number of machine learning models or tested their proposed models without evaluating how well they performed in relation to other accepted techniques. Furthermore, not much research has looked into and assessed the effectiveness of unsupervised machine learning models. Furthermore, no research has compared the effectiveness of supervised and unsupervised algorithms for identifying network attacks, including those on smart grids. Internet services are offered on demand using cloud computing technologies, with resources priced according to usage [14–15]. To capitalise on the advantages of cloud computing, government agencies and businesses have started integrating it into their operations [16–17]. This is because the technology doesn't require large investments in infrastructure or development. Cloud computing security is critical since personal and corporate data are kept in cloud data centres, which might be vulnerable to security breaches if a hacker gains access to a network [18]. Since the network serves as the system's backbone and facilitates customers' access to cloud services, any risks or vulnerabilities therein have a direct impact on the security and success of the cloud. Therefore, it is crucial to protect the network from any potential dangers. The cloud employs a variety of cybersecurity strategies, including intrusion detection systems, firewalls, and intrusion prevention systems, to address various security issues. Cloud computing must manage a multitude of assaults, including distributed denial-of-service (DDoS) attacks, DDoS attacks, SQL injection attacks, cross-site scripting (XSS) attacks, and DDoS attacks [19–20]. Repelling and identifying network threats is the main security concern for cloud computing. Inadequate countermeasures have led to an upsurge in network attacks recently; these security difficulties can be resolved with an IDS. Any IDS model must be well-established and able to support the intended workload before it can be implemented in a cloud environment. Thus, the goal of this research is to create an IDS that is equally effective.

An IDS is used to monitor all traffic and detect network intrusions in order to determine whether incoming or outgoing packets have been affected [21]. Conventional approaches for detecting intrusions use knowledge-based and statistical methodologies. These techniques had trouble evaluating vast amounts of network traffic data and identifying unknown assaults [22]. The field of machine learning has a plethora of opportunities for enhancing the security of cloud computing, the Internet of Things, and other networked systems using dependable ways. One well-known use of machine learning is in IDS development. This system's objective is to examine network data, distinguish between typical and anomalous activity, and accurately categorise it [23]. Because ML models are able to recognise intricate traffic patterns and reliably identify an assault, it is thought that ML techniques are superior to conventional models. 24]. Basic machine learning models are not as effective against more intricate and varied incursions. Studies using traditional methods show that ensemble models perform better when compared to ML-based classifiers employed alone. Ensemble-based models with reduced false alarm rates and increased accuracy are created using ML and DL models [25]. Ensemble-based models have better classification rates and data processing power. Combining automated and filter-based feature selection strategies has increased the accuracy of ensemble-based cloud infiltration detection. By removing superfluous and unnecessary features, stacked autoencoder based automatic feature selection aids in reducing the size of the feature set [26].

1.1 Based on the employed detection techniques

Depending on the detection techniques employed, IDS are broadly classified as one of two types [28]:

1. **Signature-based IDS:** Also known as IDS based on misuse. Using particular patterns—referred to as signatures in intrusion detection systems—this technique finds intrusions. This intrusion detection system makes predictions about upcoming attacks based on historical assault scenarios. The signature database

contains only known attack signatures. However, this method cannot detect new and inventive attacks, but it is effective against known ones. This IDS may overlook some potential attacks if network traffic is exceptionally high [29, 30].

2. Anamoly-based IDS: The increasing proliferation of malicious software led to the development of anomaly-based IDS, which are designed to detect unknown threats. Machine learning is used to train an anomaly-based detection system to recognise a normalised baseline, as opposed to looking for known threats. Every action on the network is compared to the baseline, which stands for the average system performance. Machine learning-based intrusion detection systems have an advantage over signature-based intrusion detection systems in that their models may be trained based on the hardware and application configurations. Zero-day attacks can be detected by AIDS, although it has a significant false-positive rate.

2. Literature Review

Bakro et al. (2023) described Public and private organizations have dramatically expanded their use of cloud computing. Attacks against cloud computing, however, significantly increase the risk to the security and privacy of data. Therefore, a reliable system for identifying cloud-based attacks is essential. When filtering techniques and automated models for feature selection are combined, superior feature sets can be obtained. The ensemble classifier is composed of numerous machine learning and deep learning models, such as an XGBoost, an extended short-term memory, a support vector machine, and a fast-learning network. To produce more accurate predictions, the weights of the suggested ensemble model are determined using the CSA. The experiments use the following three datasets: CSE-CIC-IDS-2018, Kyoto, and NSL-KDD. The simulation results demonstrate that the proposed system outperformed state-of-the-art techniques in terms of precision, recall, and F-measure. All datasets showed higher detection and false alarm rates for different types of attacks. The false positive and false negative rates of each classifier were compared to the ensemble models in order to show the ensemble model's dependability.

Talaei Khoei et al. (2022) examined Cyberattacks are becoming more sophisticated, thereby making intrusion detection more challenging. If incursions are not halted, confidence in security services, such as data privacy, availability, and integrity, may be lost. Various intrusion detection strategies for combating cybercrime have been demonstrated. Signature-based and anomaly-based IDS are the two most common classifications for these methods. The provides a classification of contemporary IDS, a review of significant recent efforts, and a listing of the most common evaluation datasets. To further the goal of making computers safer, it describes the methods that attackers use to avoid being discovered and examines the challenges of developing countermeasures.

Kumar et al. (2022) described the cloud computing paradigm has expanded over the past few years. As the number of cloud services multiplies at an exponential rate, filtering based on quality of service becomes increasingly essential. The evaluation of cloud service functionality using a variety of performance metrics also makes this a challenging endeavour. Consequently, users are confronted with a difficult and crucial task: choosing the optimal cloud services. Existing methods for selecting cloud services require users' preferences to be quantified. It's challenging for consumers to express specific preferences because of subjectivity and fuzziness. Furthermore, the current weighted summing approach does not take into account the links between QoS attributes, which leads to misleading findings because many QoS attributes are connected. We suggest a technique to choose the optimal cloud service while taking the user's preferences and QoS limitations into account in order to solve this problem.

Kumar et al. (2022) studied the methodology for rating cloud services according to the importance of quality-of-service standards in a confusing setting. The Fuzzy-AHP approach for figuring out how important each service quality criterion is in relation to the others, and the framework for choosing a cloud service. The TOPSIS technique is used in conjunction with the predetermined criterion weights to produce a general evaluation of cloud services.

Khoei et al. (2021) explained A smart grid is a new technology that uses two-way communication to intelligently distribute electricity to customers. The advantages of this network are outweighed by cyberattacks. Systems for detecting intrusions could be useful. Although the flaws of this system have been extensively studied, no workable fixes have been offered. These problems are false alarms and low detection rates. Accuracy, detection, false alarm, and missed detection rates assess performance. Three classic machine learning models were compared and contrasted: support vector machine, naïve Bayes, and K nearest neighbour. The benchmark used for all data collection is CICDDoS 2019. ReliefF gives the model training features priority. the most effective hyperparameter values for each model using the Tree-structured Parzen Estimator optimization approach. The Categorical Boosting classifier performs better in all four criteria than the three conventional models.

Bakro et al. (2021) looked at Our everyday life now wouldn't be the same without cloud computing, particularly in light of the COVID-19 pandemic & the

requirement to perform all meetings and business remotely from home. The Internet and parallel distributed/networked computing have changed the world. The future of IT will likely be shaped by cloud computing, according to current trends. The need for secure remote access to resources, apps, enormous storage, and data processing is growing. Data encryption enables secure transmission. In this study lays the theoretical groundwork for cloud computing and assesses RSA and ECC encryption methods. Our simulations show that ECC is faster than RSA.

Mighan et al. (2021) The issue of processing a large amount of security-related data for network intrusion detection is successfully addressed in this study. A data processing engine called Apache Spark is used to handle massive volumes of network traffic data. They also proposed a hybrid approach that blends deep neural networks and machine learning. After latent feature extraction, classification-based intrusion detection employs stacked autoencoder networks, random forests, decision trees, naive Bayes, and support vector machines. Large volumes of network traffic data can be quickly and effectively scanned for intrusions using classification-based intrusion detection systems. The precision, sensitivity, f-measure, and time of the approach are assessed using a real-time UNB ISCX 2012 dataset.

Mayuranathan et al. (2021) Attacks known as Distributed and Targeted Denial-of-Service pose a major risk to cloud accessibility. An operational network-layer intrusion detection system can resolve this issue. Conventional IDS on the cloud has limited detection accuracy and high processing complexity. They provide a classification method for detecting DDoS attacks based on an efficient feature subset selection. IDS feature sets are chosen using Random Harmonic Search to get the highest level of DDoS detection. Once attributes are selected, a DL-based classifier model employing Restricted Boltzmann Machines detects DDoS. The RBM's visible and hidden levels are supplemented by seven additional layers to expedite the identification of DDoS attacks. By modifying the deep RBM model's hyperparameters, precise results can be achieved. The RBM model's visible stratum has a Gaussian probability distribution. We use the KDD'99 dataset to evaluate the RHS-RBM model. The highest values of sensitivity, specificity, accuracy, F-score, and kappa for the RHS-RBM model were, in order, 99.88, 99.96, 99.92, and 99.84. The RHS-RBM model's values are superior to RBM models that do not use the RHS approach.

Islam et al. (2020) looked at Because anomaly detection systems identify hidden and unknown attacks, they are essential for locating intruders or suspicious activity. It is challenging to compare the works on this dataset with those that use the same dataset but different validation procedures because the former either used only one assessment configuration or lacked an adequate validation strategy. Using the machine learning classifier LightGBM, this dataset was classified using a binary classification scheme. Using ten-fold cross-validation, our model achieved 97.21 percent, 98.33 percent, and 96.2 percent f1_scores on the training, test, and combined datasets, respectively. In addition, a separate set of test data resulted in a f1_score of 92.96 percent for a model fitted with only train data. Therefore, our model performs remarkably well with unknown data.

Khraisat et al. (2019) described Cyberattacks are becoming increasingly sophisticated, which makes intrusion detection increasingly difficult. The trust of security services such data confidentiality, availability, and integrity may be jeopardised if the incursions are not stopped. A variety of intrusion detection techniques have been put out to address threats to computer security. These methods can be divided into two main categories: intrusion detection systems that rely on signatures and those that use anomaly-based methods. The offers an overview of frequently used datasets for evaluation purposes, a thorough survey of noteworthy recent efforts, and a taxonomy of modern IDS.

Sharma et al. (2018) looked at One kind of mobile ad hoc network that has many excellent uses in the intelligent traffic system is the vehicular ad hoc network. Since there is a risk to human life when using VANETs, it is imperative to develop the safest possible interaction between vehicle nodes. Numerous security measures are designed to protect VANETs, with intrusion detection systems being the most prevalent. Because of their distinct features—such as resource-constrained nodes, high node mobility, particular protocol stacks, and standards—implementing IDS in VANET-like networks is difficult. IDS has already proven to be useful in traditional network detection of malicious nodes. Furthermore, basic guidelines for creating IDSs with prospective VANET and VANET Cloud applications have been supplied. Our objective is to pinpoint important trends, unresolved issues, and future research areas related to IDS deployment in VANETs.

Table 1. Comparison of review table

Reference	Topic	Methodology/Approach	Findings
Bakro et al. (2023)	Cloud Intrusion Detection	Ensemble Classifier with CSA	- Enhanced feature sets combining filter and automated models - Ensemble model with LSTM, SVM, XGBoost, and FLN - Achieved greater accuracy, recall, precision, and F-measure
Talaei Khoei et al. (2022)	Intrusion Detection Methods	Taxonomy and Review	- Categorized IDS into SIDS and AIDS - Overview of commonly used datasets and evasion techniques
Kumar et al. (2022)	Cloud Service Selection	Fuzzy-AHP and TOPSIS	- Proposed framework considering user preferences and QoS - Improved cloud service selection decision-making
Kumar et al. (2022)	Cloud Service Evaluation	Fuzzy-AHP and TOPSIS	- Architecture for cloud service selection process - Calculated weights of QoS criteria and ranked cloud services
Khoei et al. (2021)	Smart Grid Intrusion Detection	Boosting-based Models	- Comparison of boosting-based models with traditional models - Categorical Boosting achieved highest performance metrics
Bakro et al. (2021)	Cloud Computing Security	Simulation and Analysis	- ECC outperformed RSA in encryption algorithms
Mighan et al. (2021)	Network Intrusion Detection	Apache Spark and Hybrid Method	- Used Apache Spark for processing voluminous network traffic - Hybrid method with deep neural networks and ML techniques
Mayuranathan et al. (2021)	DDoS Detection	RHS Optimization and Deep RBM	- Optimal feature selection with RHS and Deep RBM - High detection

			performance on KDD'99 dataset
Islam et al. (2020)	Anomaly Detection	Machine Learning with LightGBM	- Achieved high f1_scores on train, test, and combined datasets - Demonstrated substantial performance on unknown data
Khraisat et al. (2019)	Intrusion Detection Methods	Literature Review	- Provided taxonomy and overview of IDS and datasets
Sharma et al. (2018)	VANET Security	IDS Deployment in VANET	- Identified challenges and future research directions

3. Research Methodology

The system block diagram for the proposed method to detect intrusion attacks is depicted in Figure 1. We test our model on the CICDDoS2019 dataset. Accuracy, f-measure, precision, recall, and performance are the metrics used to assess our suggested approach.

Data collection, data pre-processing, a classification machine model, and an assessment to ascertain whether the attack qualifies as a DDoS are the four main parts.

In the initial phase of data acquisition for the proposed model, the pre-processing dataset is emphasised. IP traffic data are input and cleaned data are output in the second stage of pre-processing. In order to maximise the effectiveness and efficiency of the training process, data preparation is an essential stage in the deep learning process. Feature selection by removing irrelevant data, absent value management, label conversion, categorization, and data normalization are included. However, not all features contribute equally to DDoS attack detection. After evaluating the suggested approach, the model suggested a hybrid deep learning model that blends XceptionNet+2D-CNN to classify network traffic as benign or normal, thus facilitating intrusion categorization of real-time data traffic. 2D-CNN architectures' parameters are optimised with the application of Ant Colony Optimisation.

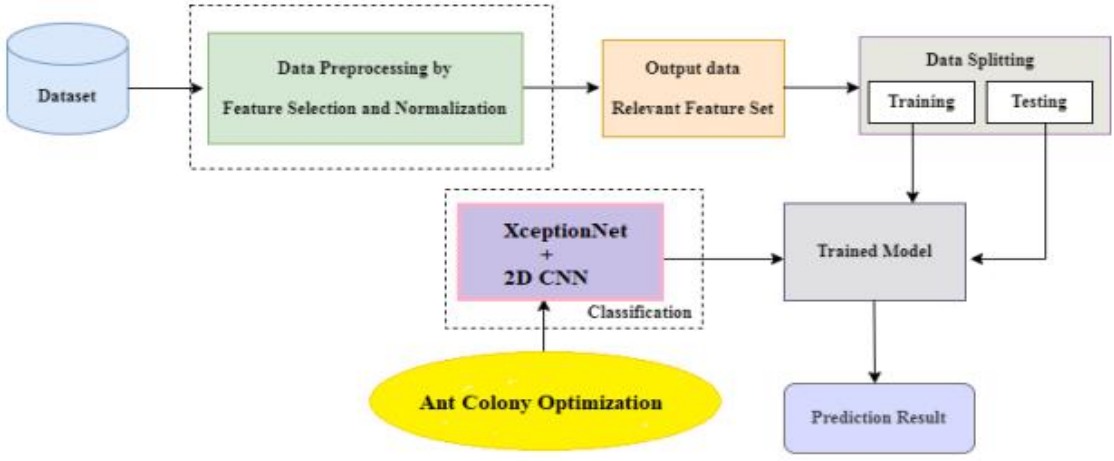

Figure 1: Proposed Methodology

- **Data Acquisition:** The pre-processing dataset is highlighted in the first stage of the proposed model's data acquisition. IP traffic data are input during pre-processing and cleaned data are output during the second step.
- **Data pre-processing:** Because it has the potential to significantly increase the effectiveness and efficiency of the training process, it is an essential duty in deep learning. It includes selecting features by removing unnecessary data, missing value management, label conversion, categorization, and data normalization. However, not all features contribute equally to the detection of DDoS attacks. The proposed model concludes by proposing a hybrid deep learning model

that incorporates Resnet50+3D-CNN for intrusion classification of real-time data traffic and evaluating the proposed approach. Utilizing Black Widow Optimization, the parameters of 3D-CNN architectures are optimized.

- **Classification Machine Model:** The component refers to a machine learning model's genuine classification of cyberattacks. It could be one of the classification models previously mentioned, such as Random Forest, SVM, or Neural Networks. The model is trained on a labelled dataset, where the input features are the pre-processed data and the output is the predicted class indicating whether or not the attack is a DDoS attack.

1. Evaluation To Produce the Output of Whether the Attack Is Ddos: Many assessment criteria based on the classification model's predictions can be used to identify whether an attack is a distributed denial of service attack. The evaluation component assesses the classification model's capacity to detect DDoS attacks. putting into practise appropriate evaluation metrics, as previously mentioned, such as recall, precision, F1 score, or AUC-ROC. The assessment process makes it easier to determine how well the model performs and highlights its advantages and disadvantages. The evaluation's output is a determination of whether or not the attack qualifies as a DDoS. A few popular assessment measures for binary classification issues are as follows:

2. Accuracy: It computes the ratio of instances correctly classified as DDoS attacks or non-DDoS attacks to the total number of instances to ascertain the classification model's overall accuracy. Nonetheless, if the dataset is unbalanced, precision can be deceiving.

Accuracy = (Number of correct predictions) / (Total number of predictions)

3. Precision: The ratio of correctly identified DDoS attacks as true positives to the total number of false positives and true positives that were incorrectly classified as DDoS attacks is known as precision. Precision is useful in reducing false positives and concentrates on the accuracy of affirmative forecasts.

Precision = True Positives / (True Positives + False Positives)

4. Recall Sensitivity or True Positive Rate: Recall calculates the ratio of true positives to the sum of true positives and false negatives for DDoS attacks that are misidentified.. It assesses how well the model can detect every positive instance. A high recall rate suggests that the model can identify DDoS attacks.

5. F1 Score: Recall and precision are averaged to get the F1 score. It offers a thorough assessment of the model's effectiveness by taking recall and precision into account. where there is an uneven distribution of classes or where false positives and false negatives have similar relevance, it is beneficial.

F1 Score = 2 * (Precision * Recall) / (Precision + Recall)

1.2 Machine learning models

- **Naïve Bayes algorithm**

1. Input: A labelled dataset, with each data point consisting of a set of features (X) and a corresponding class label (Y), is fed into the algorithm.

2. Training: To begin, the algorithm determines the prior probability P(Y) for every class label Y. To achieve this, divide the total number of data points by the number of times each class label appears in the training dataset.

3. Feature Likelihoods: For each feature x_i and each class label Y, the algorithm calculates the likelihood probability $P(x_i \mid Y)$. This is done by counting the occurrences of feature x_i in data points belonging to class Y and dividing by the total number of data points in class Y. Laplace smoothing is often applied to avoid zero probabilities for unseen features.

4. Prediction: The technique forecasts the class label Y for a new data point with features by using the Bayes theorem to find the posterior probability P(Y | x) for each class label Y. x:P(Y | x) = P(x | Y) * P(Y) / P(x) In this case, P(Y) is the prior probability of class Y, P(x) is a normalisation factor, and P(x | Y) is the product of the likelihood probabilities for each feature x_i given y.

Classification: The projected class for the new data point is chosen by the algorithm to be the class label Y with the highest posterior probability.

Output: The algorithm returns the predicted class labels for all new data points.

The Nave Bayes algorithm is straightforward, efficient, and effective for text classification, spam filtering, and other classification tasks, particularly when the independence assumption is approximately true. It is important to note that while Nave Bayes can function well for certain types of data, it may not be suitable for highly correlated or complex datasets where the independence assumption is not valid.

- **Logistic Regression algorithm**

1. Input: The algorithm takes a labelled dataset as input, where each data point consists of a set of features (X) and a binary class label (Y) indicating one of two classes (e.g., 0 or 1).

2. Data Preparation: If needed, feature scaling and data normalization can be performed to ensure that all features have a similar scale, which can improve the convergence and performance of the algorithm.

3. Model Initialization: Initialize the model parameters (weights) and bias term to small random values or zeros. The model aims to learn the best weights that minimize the error in predicting the class labels.

4. **Hypothesis Function:** A hypothesis function is used by the Logistic Regression model to forecast the likelihood that a data point is a member of class 1. The definition of the hypothesis function is:

$$h\theta(x) = \sigma(\theta^T * x + b)$$

where $h\theta(x)$ is the predicted probability, σ is the sigmoid function (activation function), θ represents the weight vector, x is the feature vector, and b is the bias term.
The sigmoid function $\sigma(z)$ is defined as:

$$\sigma(z) = 1 / (1 + e^{-z})$$

5. **Cost Function:** The approach uses a cost function, also called a loss function, to measure the error between the predicted probability and the actual class labels. In logistic regression, the cost function that is most frequently utilised is the cross-entropy (log loss) function.

$$J(\theta, b) = -1/m * \Sigma [y * \log(h\theta(x)) + (1 - y) * \log(1 - h\theta(x))]$$

where m is the number of data points, Σ represents the sum over all data points, y is the actual class label, and $h\theta(x)$ is the predicted probability.

6. **Gradient Descent:** Finding the ideal values for the model parameters (θ and b) in order to minimise the cost function is the algorithm's main objective. To get the lowest cost, Gradient Descent is utilised to iteratively update the parameters in the direction of the steepest descent.

7. **Training:** During the training phase, the algorithm performs multiple iterations, updating the model parameters using Gradient Descent until convergence or after a fixed number of epochs.

8. **Prediction:** The model can be used to predict the class label for fresh data points once it has been trained. The predicted class label is determined by comparing the predicted probability $h\theta(x)$ with a threshold often 0.5. If $h\theta(x) >= 0.5$, the predicted class is 1; otherwise, it is 0.

9. **Output:** The Logistic Regression algorithm outputs the learned model parameters weights and bias and can be used for binary classification tasks.

Logistic Regression is a popular classification algorithm in machine learning due to its simplicity, effectiveness, and interpretability. It works well for binary classification jobs and may be expanded to multiclass classification using methods like Softmax Regression and One-vs-Rest (OvR).

- **SVM algorithm**

1. **Input:** The algorithm takes a labelled dataset as input, where each data point consists of a set of features (X) and a corresponding binary class label (Y) indicating one of two classes (e.g., 0 or 1).

2. **Data Preparation:** If needed, feature scaling and data normalization can be performed to ensure that all features have a similar scale, which can improve the convergence and performance of the algorithm.

3. **Feature Space Transformation Kernel Trick - Optional:** SVM can use a kernel function to transform the original feature space into a higher-dimensional space. This allows SVM to find a linear decision boundary in the transformed space, which may be nonlinear in the original space. Common kernel functions include the linear kernel, polynomial kernel, radial basis function RBF kernel, and sigmoid kernel.

4. **Margin Maximization:** The main objective of SVM is to find the optimal hyperplane that maximizes the margin between the two classes. The margin is the distance between the hyperplane and the closest data points of each class support vectors. The optimal hyperplane is the one that achieves the largest margin while correctly classifying the data points.

5. **Soft Margin Optional**: SVM introduces slack variables to allow for a soft margin in situations where the data is not perfectly separable. Certain data points may be misclassified or fall inside the margin zone thanks to these slack factors, but misclassification will result in a penalty. Finding a balance between classification error and margin maximisation is the goal.

6. **Optimization Problem:** The SVM technique frames the issue as a convex optimisation problem in order to determine the ideal weights and biases for the hyperplane. The optimisation seeks to maximise the margin and minimise the misclassified data points' classification error.

7. **Kernel Parameters Optional:** In the event that a kernel function is employed, the SVM algorithm may further entail fine-tuning the kernel parameters, such as the polynomial kernel's degree or the RBF kernel's width, in order to maximise performance on the training set.

8. **Training:** During the training phase, the algorithm uses techniques like sequential minimal optimisation (SMO) or quadratic programming to solve the optimisation problem and determine the optimal hyperplane parameters.

9. **Prediction:** Once trained, the model can be used to predict the class label for new data points. The predicted class label can be determined by comparing the signed distance between the hyperplane and the data point. If the signed distance is positive, the data point is assigned to one class; if not, it is assigned to the other class.

10. **Output:** For binary classification problems, the Support Vector Machine approach can be used to output the learnt model parameters, which include weights and bias.

SVM is a robust classification algorithm that is renowned for its capacity to manage high-dimensional data and locate complex decision boundaries. It is particularly useful when the data are distinct and the classes are well-defined. Using techniques such as One-

vs-One (OvO) or One-vs-Rest OvR strategies, SVM can be adapted to perform multiclass classification. In addition, the Support Vector Regression SVR variant of SVM can be used for regression assignments.

- **AdaBoost algorithm**

1. Input: The algorithm takes a labelled dataset as input, where each data point consists of a set of features X and a corresponding binary class label Y indicating one of two classes (e.g., 0 or 1).

2. Weight Initialization: Initially, each data point is assigned an equal weight, making them equally important for the first round of training.

3. Training Rounds Iterations: The AdaBoost algorithm performs a series of training rounds, often referred to as "iterations" or "weak learners."

4. Weak Learner Selection: A "weak learner," or a straightforward classifier that outperforms random guessing by a tiny margin, is trained on the dataset for each iteration. Small decision trees with little depth or decision stumps decision trees with a single split are typical instances of weak learners.

5. Weighted Training: During training, the algorithm assigns higher weights to misclassified data points, making them more important for the next round of training. This emphasizes the misclassified data points, forcing the weak learner to focus on correcting the errors.

6. Weak Learner Weight Calculation: Each weak learner is assigned a weight based on its accuracy in classifying the data points. More accurate weak learners receive higher weights, indicating that they are more influential in the final ensemble.

7. Ensemble Creation: After each iteration, the weak learner's performance and weight are recorded, and it becomes part of the ensemble model.

8. Weighted Voting: During prediction, the AdaBoost algorithm combines the predictions of all weak learners, each weighted by its corresponding weight, to make the final prediction. More accurate weak learners are guaranteed a larger voice in the final forecast thanks to the weighted voting.

9. Final Prediction: The weighted total of all the predictions made by the poor learners forms the basis of the final forecast. For the new data point, the predicted class label is the class with the highest weighted vote.

10. Output: The ensemble of weak learners and the weights that go along with them are produced by the AdaBoost method and can be used to predict the class labels of fresh data points.Using the strengths of several weak learners, AdaBoost is an ensemble learning technique that generates a robust classifier. It adapts its focus on misclassified data points iteratively, resulting in enhanced performance over time. AdaBoost excels at dealing with complex datasets and achieving high accuracy, even when using feeble classifiers. However, it can be sensitive to chaotic data and outliers, and careful parameter tuning may be required to prevent overfitting.

1.3 Machine Learning-Based Approaches for Implementing Intrusion Detection Systems

Enhanced intrusion detection systems (IDS) have become a necessity due to the proliferation of malicious software. As a result, an effective intrusion detection system (IDS) capable of detecting novel, sophisticated malware is required. In the past several decades, machine learning has been used to enhance intrusion detection. The process of extracting knowledge from vast quantities of data is known as machine learning. Models of machine learning are comprised of a set of rules, methods, or complex "transfer functions" that can be used to recognize and predict behaviour, as well as intriguing data patterns. IDS has seen extensive application of machine learning techniques [31]. In a machine-learning-based approach, a model of analyzed patterns is created and periodically updated to enhance the performance of anomaly detection. The objective of employing machine-learning techniques is to generate IDSs that are more accurate and require less human understanding. The primary emphasis of IDS research based on machine learning is the discovery of patterns and the development of intrusion detection systems based on the dataset. It consists of both Classification and Ensemble-learning based models.

4. Result

The current CICDDoS techniques, such as the gradient boosting classifier and the Adaboost classifier, are compared with one another in order to evaluate the hybrid ML-F method. Table 2 compares the outcomes of the binary classification for CICDDoS in the proposed hybrid ML-F with the current methods. Table 1 displays the accuracy, precision, recall, and f-measure of the binary classification result for the CICDDoS dataset. The gradient boosting classifier and the adaboost classifier are two current methods that are compared with the suggested hybrid ML-F methodology. Using the CICDDoS datasets, the suggested technique outperforms the others in binary classification. With the suggested hybrid ML-F technique, accuracy Naïve Byes 0.69 f1 0.66, precision 0.69, recall 0.69, were attained. The Logistic Regression With an accuracy of 0.86%, precision of 0.86%, recall of 0.86%, and f-measure of 0.86%, this gradient boosting classifier performed admirably. In SVM terms, At the moment, the classifier's f-measure was 0.86%, accuracy was 0.86%, and average recall was 0.86%. Visual results are displayed for CICDDoS binary classification using the present approach. The binary classification results for the CICDDoS dataset are displayed in Table 2 along with the error rate and AUC score.

International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Table 2 displays the accuracy, precision, recall, and f-measure of the binary classification result of the CICDDoS dataset.

Accuracy	0.689739333559733
F1-score	0.6794101347617421
precision	0.7056274548734854
Recall	0.689739333559733

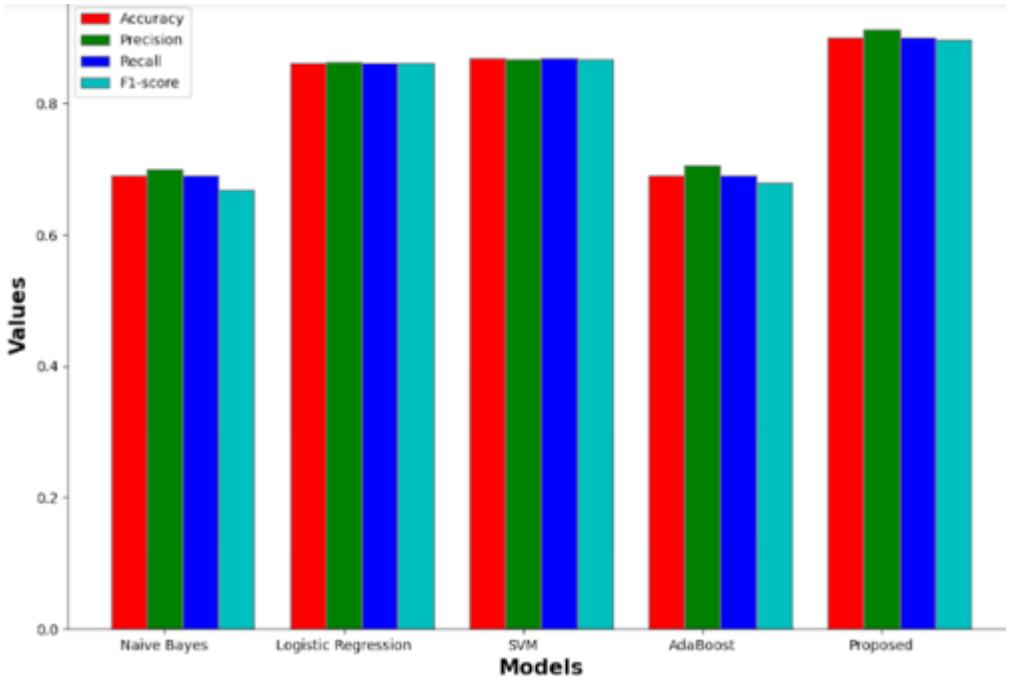

Fig 2: - The Results of Graphical Representation**.**

Table 3: displays the binary classification outcomes for the CICDDoS dataset**.**

	Model	**Accuracy**	**F1-score**	**precision**	**Recall**
0	Naïve Bayes	0.690335	0.668135	0.699899	0.690355
1	Logistic Regression	0.861980	0.861965	0.862249	0.861980
2	SVM	0.868810	0.868212	0.868428	0.868810
3	AdaBoost	0.689739	0.679410	0.705627	0.689739
4	Proposed	0.900621	0.896912	0.912692	0.900621

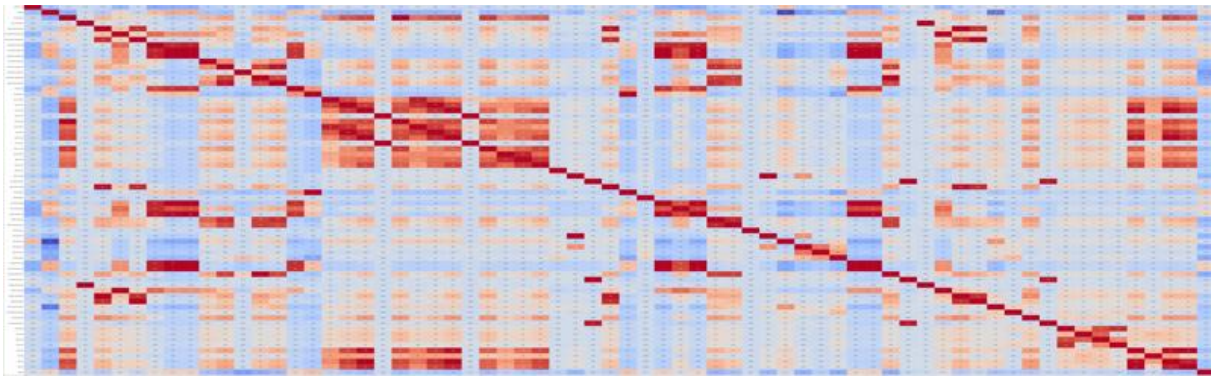

Fig 3: Graphical represention measure of the model's performance.

Accuracy: The accuracy indicates how effectively the model classified the cases overall with precision. The accuracy in this instance is 0.86, indicating that the model successfully classified 86.20% of the cases.F1-score: The harmonic mean of precision and recall is used to calculate this balanced measure of the model's performance. The model appears to do reasonably well overall in terms of precision and recall, based on your reported F1-score of 0.86. Precision is the percentage of accurately categorised positive instances among all instances that were projected to be positive. With a precision value of 0.86, you have indicated that 86.22% of the occurrences that were anticipated to be positive were, in fact, positive.Recall: Recall, also known as sensitivity or true positive rate, quantifies the proportion of correctly classified positive events among all real positive instances. Your provided recall value of 0.86 indicates that the model correctly detected around 86.20% of the actual positive events. All things considered, the model appears to have performed well, exhibiting high levels of recall, accuracy, and precision. However, in order to determine whether these metrics satisfy the intended needs or if they may be improved, it's crucial to take the context and the particular problem domain into account.

Table 4. comparision betweeen Accuracy , F1 score , Precision , Recall requirements.

Accuracy	0.8619797700244702
F1-score	0.8619649618801627
precision	0.8622492846787924
Recall	0.8619797700244702

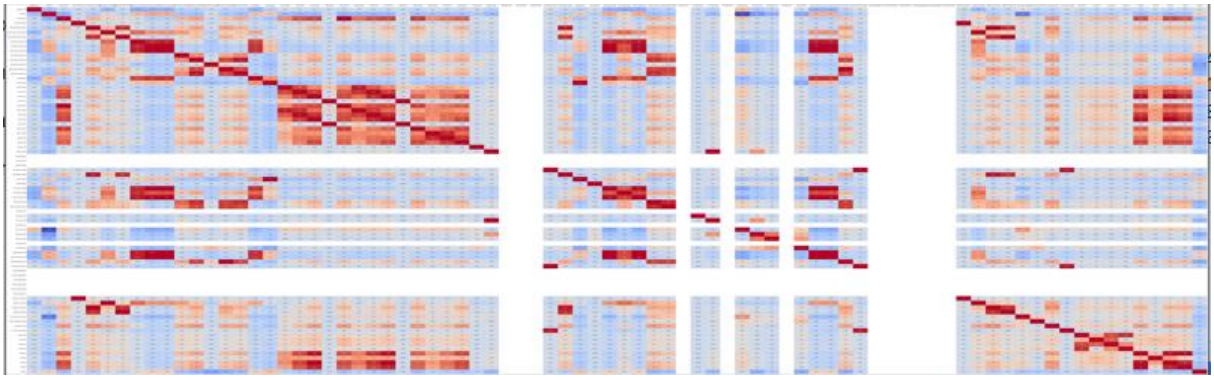

Fig:-4 The graphical represention good performance with high accuracy, precision, and recall**.**

5. Conclusion

Using LR techniques, two new anomaly-based intrusion detection models were developed. In both binary class and multiclass classification problems, models based on LR perform well, according to analysis and experimental results on the NSLKDD benchmark dataset. They perform similarly to SVM-based intrusion detection models, but better than the Naive Bayes-based model. However, LR-based models are more ideal for use in real-time network monitoring and intrusion detection analyses as they have significantly lower processing overhead than SVM-based models**.** In this research Using blockchain to safeguard data analytics and

decision-making through IoT technologies. Recently, blockchain-based FL designs have demonstrated potential for resolving the security and privacy concerns posed by data analytics in IoT applications. However, these architectures may introduce new security, privacy, and efficiency concerns. In this study blockchain to demonstrate the benefits of combining the two technologies to develop secure, efficient, and dependable IoT applications. The three novel deep learning models as viable methods for detecting distributed denial of service DDoS attacks.The results of the experiment show that there is no such thing as a free lunch, which validates a well-established hypothesis. According to the findings of our investigation, there are three distinct models, and the manner in which each one operates is distinct based on the kind of DDoS attack. If you utilize strategies like feature reduction or compression, it's likely that your F1 score will drop for some attack classes. because these strategies reduce the number of features an attack class uses. This tactic has the potential to provide a considerable improvement in performance in comparison to other types of attacks. Following the implementation of Auto Encoder, the F1-score for SNMP increased from 0.68% to 70%, demonstrating a notable rise in quality. When considering UDP, the differentiation may be seen much more plainly. Because the F1 score is only 0.68%, it may be deduced that the model incorrectly classified practically all attacks as being harmless. The F1 score is now at 0.86 percent after the recent improvement. An additional illustration is provided by WebDDoS: neither DNN nor DNN combined with Auto Encoder were successful in achieving a high F1-score. The F1-score has greatly increased, and it is currently at 0.86%. This is a significant improvement when compared to the third method, which makes use of pandas-profiling. Due to the fact that authorized network users can initiate and reflect distributed denial of service attacks, it is challenging for victims to identify and prevent DDoS attacks. In this study employs the CICDoS2019 dataset as both the training and testing set in order to investigate how difficult it is to detect DDoS attacks, particularly those that have become increasingly popular in recent years.

References

[1] Islam, M. K., Hridi, P., Hossain, M. S., & Narman, H. S. (2020, November). Network anomaly detection using lightgbm: A gradient boosting classifier. In 2020 30th International Telecommunication Networks and Applications Conference (ITNAC) (pp. 1-7). IEEE

[2] Ren, J., Guo, J., Qian, W., Yuan, H., Hao, X., & Jingjing, H. (2019). Building an effective intrusion detection system by using hybrid data optimization based on machine learning algorithms. *Security and communication networks*, *2019*.

[3] Sharma, S., & Kaushik, B. (2019). A survey on internet of vehicles: Applications, security issues & solutions. Vehicular Communications, 20, 100182.

[4] Sharma, S., & Kaul, A. (2021). VANETs cloud: architecture, applications, challenges, and issues. *Archives of Computational Methods in Engineering*, *28*, 2081-2102.

[5] Axelsson, S. (1998). Research in intrusion-detection systems: A survey (Vol. 120). Technical report 98–17. Department of Computer Engineering, Chalmers University of Technology.

[6] Bace, R. G., & Mell, P. (2001). Intrusion detection systems.

[7] Smadi, A. A., Ajao, B. T., Johnson, B. K., Lei, H., Chakhchoukh, Y., & Abu Al-Haija, Q. (2021). A Comprehensive survey on cyber-physical smart grid testbed architectures: Requirements and challenges. *Electronics*, *10*(9), 1043.

[8] Tazi, K., Abdi, F., & Abbou, M. F. (2015, December). Review on cyber-physical security of the smart grid: Attacks and defense mechanisms. In 2015 3rd International Renewable and Sustainable Energy Conference (IRSEC) (pp. 1-6). IEEE.

[9] Khoei, T. T., Aissou, G., Hu, W. C., & Kaabouch, N. (2021, May). Ensemble learning methods for anomaly intrusion detection system in smart grid. In 2021 IEEE international conference on electro information technology (EIT) (pp. 129-135). IEEE.

[10] Khoei, T. T., Ismail, S., & Kaabouch, N. (2021, December). Boosting-based models with tree-structured parzen estimator optimization to detect intrusion attacks on smart grid. In 2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON) (pp. 0165-0170). IEEE.

[11] El Mrabet, Z., El Ghazi, H., & Kaabouch, N. (2019, May). A performance comparison of data mining algorithms-based intrusion detection system for smart grid. In 2019 IEEE International Conference on Electro Information Technology (EIT) (pp. 298-303). IEEE.

[12] Anthi, E., Williams, L., Słowińska, M., Theodorakopoulos, G., & Burnap, P. (2019). A supervised intrusion detection system for smart home IoT devices. IEEE Internet of Things Journal, 6(5), 9042-9053.

[13] Talaei Khoei, T., Ismail, S., Shamaileh, K. A., Devabhaktuni, V. K., & Kaabouch, N. (2022). Impact of Dataset and Model Parameters on Machine Learning Performance for the Detection of GPS Spoofing Attacks on Unmanned Aerial Vehicles. Applied Sciences, 13(1), 383.

[14] Kumar, R. R., Tomar, A., Shameem, M., & Alam, M. N. (2022). Optcloud: An optimal cloud service selection framework using QoS correlation lens. Computational Intelligence and Neuroscience, 2022.

[15] Kumar, R. R., Shameem, M., Khanam, R., & Kumar, C. (2018, December). A hybrid evaluation framework for QoS based service selection and ranking in cloud environment. In 2018 15th IEEE India council international conference (INDICON) (pp. 1-6). IEEE.

[16] Kumar, R. R., Shameem, M., & Kumar, C. (2022). A computational framework for ranking prediction of cloud services under fuzzy environment. Enterprise information systems, 16(1), 167-187.

[17] Akbar, M. A., Shameem, M., Mahmood, S., Alsanad, A., & Gumaei, A. (2020). Prioritization based taxonomy of cloud-based outsource software development challenges: Fuzzy AHP analysis. Applied Soft Computing, 95, 106557.

[18] Bakro, M., Bisoy, S. K., Patel, A. K., & Naal, M. A. (2021). Performance analysis of cloud computing encryption algorithms. In Advances in Intelligent Computing and Communication: Proceedings of ICAC 2020 (pp. 357-367). Springer Singapore.

[19] Bakro, M., Kumar, R. R., Alabrah, A. A., Ashraf, Z., Bisoy, S. K., Parveen, N., ... & Abdelsalam, A. (2023). Efficient Intrusion Detection System in the Cloud Using Fusion Feature Selection Approaches and an Ensemble Classifier. Electronics, 12(11), 2427.

[20] Bakro, M., Bisoy, S. K., Patel, A. K., & Naal, M. A. (2022). Hybrid blockchain-enabled security in cloud storage infrastructure using ECC and AES algorithms. In Blockchain based Internet of Things (pp. 139-170). Singapore: Springer Singapore.

[21] Srilatha, D., & Shyam, G. K. (2021). Cloud-based intrusion detection using kernel fuzzy clustering and optimal type-2 fuzzy neural network. Cluster Computing, 24(3), 2657-2672.

[22] Xu, C., Shen, J., Du, X., & Zhang, F. (2018). An intrusion detection system using a deep neural network with gated recurrent units. IEEE Access, 6, 48697-48707.

[23] Abbas, G., Mehmood, A., Carsten, M., Epiphaniou, G., & Lloret, J. (2022). Safety, Security and Privacy in Machine Learning Based Internet of Things. Journal of Sensor and Actuator Networks, 11(3), 38.

[24] Mighan, S. N., & Kahani, M. (2021). A novel scalable intrusion detection system based on deep learning. International Journal of Information Security, 20, 387-403.

[25] Mayuranathan, M., Murugan, M., & Dhanakoti, V. (2021). Best features-based intrusion detection system by RBM model for detecting DDoS in cloud environment. Journal of Ambient Intelligence and Humanized Computing, 12, 3609-3619.

[26] Arora, N., & Kaur, P. D. (2020). A Bolasso based consistent feature selection enabled random forest classification algorithm: An application to credit risk assessment. Applied Soft Computing, 86, 105936.

[27] Singh, S., Sharma, S., Sharma, S., Alfarraj, O., Yoon, B., & Tolba, A. (2021). Intrusion Detection System-Based Security Mechanism for Vehicular Ad-Hoc Networks for Industrial IoT. IEEE Consumer Electronics Magazine, 11(6), 83-92.

[28] Butun, I., Ra, I. H., & Sankar, R. (2015). An intrusion detection system based on multi-level clustering for hierarchical wireless sensor networks. Sensors, 15(11), 28960-28978.

[29] Sharma, S., & Kaul, A. (2018). A survey on Intrusion Detection Systems and Honeypot based proactive security mechanisms in VANETs and VANET Cloud. Vehicular communications, 12, 138-164.

[30] Sharma, S., & Kaul, A. (2018). Hybrid fuzzy multi-criteria decision making based multi cluster head dolphin swarm optimized IDS for VANET. Vehicular Communications, 12, 23-38.

[31] Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity, 2(1), 1-22.

[32] Mrs. Monika Soni. (2015). Design and Analysis of Single Ended Low Noise Amplifier. International Journal of New Practices in Management and Engineering, 4(01), 01 - 06. Retrieved from http://ijnpme.org/index.php/IJNPME/article/view/33

[33] Sivakumar, D. S. (2021). Clustering and Optimization Based on Hybrid Artificial Bee Colony and Differential Evolution Algorithm in Big Data. Research Journal of Computer Systems and Engineering, 2(1), 23:27. Retrieved from https://technicaljournals.org/RJCSE/index.php/journal/article/view/15