Self-Organizing Computational System for Network Anomaly Exploration using Learning Algorithms

[1]Preethi P, [2]Lalitha K and [3]Yogapriya J
[1,2,3]Department of Computer Science and Engineering, Kongunadu College of Engineering and Technology, Trichy, India.
[1]preethi1.infotech@gmail.com, [2]lalithabtechme@gmail.com, [3]yogapriya.j@gmail.com

Correspondence should be addressed to Preethi P : preethi1.infotech@gmail.com

Article Info
Journal of Machine and Comp uting (http://anapub.co.ke/journals/jmc/jmc.html)
Doi: https://doi.org/10.53759/7669/jmc202303035
Received 28 February 2023; Revised from 12 June 2023; Accepted 06 July 2023.
Available online 05 October 2023.

Abstract – The forum in the nation for reporting information security flaws had 14,871 reports by the end of 2021, a 46.6% increase from 2020. The total of 5,567 high risk vulnerabilities, an increase of nearly 1,400 over the previous year. Evidently, both the total number of vulnerabilities found annually, and the total number of high-risk vulnerabilities are rising. In order for data mining technology to play a wider part in the predictive investigation of network security models, it is advised that its capability have to be improved. This paper combines the concepts of data mining (DM) with machine learning (ML), which introduces similar technologies from DM technology and security establishing collection channel, thereby finally introduces the computer network security maintenance process based on data mining in order to improve the application effect of DM in the predictive analysis of network security models. In this paper, a self-organizing neural network technique that detects denial of service (DOS) in complicated networks quickly, effectively, and precisely is introduced. It also analyses a number of frequently employed computer data mining methods, including association, clustering, classification, neural networks, regression, and web data mining. Finally, it introduces a computer data mining method based on the self-organizing (SO) algorithm. In comparison to conventional techniques, the SO algorithm-based computer data mining technology is also used in defensive detection tests against Dos attacks. A detection average accuracy rate of more than 98.56% and a detection average efficiency gain of more than 20% are demonstrated by experimental data to demonstrate that tests based on the Data Mining connected SO algorithm have superior defensive detection effects than standard algorithms.

Keyword – Self-Organizing, Geometric Neighborhood, Regression, NSL-KDD, Over-Sampling, Under-Sampling and Data Imbalance.

I. INTRODUCTION

Cyberattack risk has increased dramatically due to the exponential growth of computer applications and network sizes [1]. A network can change, thus it's crucial to ensure sure it's safe [2]. Using a powerful collection of tools, attackers can alter the computer system. To thwart potential intruders, it is essential to predict network incursions and identify the type of assault that has been launched [3]. A model to predict cybersecurity issues has been proposed by academic institutions and businesses using reports, network activity, social media, and website data [4]. The creation of an autonomous detection mechanism that will assist in promptly identifying and classifying the various types of intrusions at the particular network infrastructure stages [5] is a proactive way to anticipate cybersecurity.

The bulk of security databases are unbalanced due to the frequency of cyberattacks. The unbalanced data makes it difficult to categorise cyberattacks precisely. For supervised machine learning algorithms to forecast accurately, a lot of data is required. Unbalanced datasets can be handled using algorithms like oversampling, undersampling, and balancing sampling. Minority class detection is typically greatly enhanced by oversampling approaches, which combine samples from the minority classes to produce a balanced range of data. Contrarily, undersampling techniques remove samples from the classification model to create a more accurate representation. Undersampling techniques are prone to overfitting, nevertheless, because they could leave out certain crucial instances of the majority class that are necessary to distinguish between the majority and minority classes. To achieve balancing sampling, oversampling and undersampling techniques are employed.

The majority class typically performs worse when using any of these tactics. The biggest issue with an unbalanced dataset is maintaining the majority class' effectiveness while enhancing threat recognition. Additionally, if cyberattack alerts are sent out too frequently, people start to tune them out and ignore them. One of the best performance indicators for IDS products is the weighted F1 score.

Different supervised machine learning techniques were evaluated on the benchmark network dataset known as NSLKDD [6] We compared various deep learning and machine learning algorithms. A hybrid voting classifier is also employed for better outcomes. The job's presentation involves five different classifications. Resampling was used to assess every classifier that was selected. The SMOTE family contains oversampling. Near Miss is the name of the undersampling method that is employed. SMOTE and Edited Nearest Neighbour (SMOTEENN), a mixture of SMOTE and ENN, is the final balancing sampling approach taken into consideration in this study. The highlights of the proposed work are,

- This research pioneers the integration of data mining (DM) and machine learning (ML) concepts to fortify the predictive investigation of network security models. This cohesive approach enhances the ability to detect and mitigate potential threats within complex network environments.
- The paper introduces a cutting-edge technique known as the Self-Organizing Neural Network (SONN), designed to expedite the detection of denial of service (DOS) attacks.
- The effectiveness of the Data Mining-connected SONN algorithm is validated through extensive experimentation. Results showcase an exceptional average detection accuracy rate exceeding 98.56%.
- SONN model can swiftly pinpoint anomalous behavior indicative of DOS attacks, ensuring timely threat identification even if the network is large-scale or highly dynamic.

The remaining sections are organized as: In Section 2, the research that has already been conducted on anomaly detection is given. The proposed Self-organising learning paradigm for Anomaly Exploration was explained in the third section. The suggested method's experimental findings on a dataset that is openly available are explained in the fourth section, which also highlights the value of further study in this field and potential effects of the work. The paper is then concluded in the fifth section with references.

II. RELATED WORKS

By considering how successfully neural networks outperformed intrusion detection systems, Deep Neural Network was developed by [7]. They gave an example of how to thoroughly analyse the DNN employing types of hidden layers and benchmark network traffic sources. Their proposed strategy outperformed currently employed traditional machine learning methods. Yet another investigation [8] that employed the NSL-KDD database and a recurrent neural network found that it was 81% efficient in multi-class classification. The NSLKDD database was made into an image file and applied to CNN model to be able to enhance prediction performance. According to [9], there was a slight improvement in the categorizing results.

Additionally, it was shown that auto encoder systems surpass the commonly used technique for dimension reduction known as principal component analysis in terms of dimensionality reduction [10]. [Reference required] High-level data from the NSLKDD database was obtained by [11] using an auto-encoder, which outperformed categories like KNN and SVM in terms of accuracy and false positive rate (SVM). [12] proposed a conditional variational auto-encoder with an 80% reliability rate for the mechanism for detecting network intrusions.

The author of [13] described SMOTE and adaptable synthetic samples on the NSLKDD dataset. It was determined that their suggested generative discretization auto-encoder oversampling methodology is superior to currently used oversampling methods that make use of classification tools including RF, linear SVM, Regression models, and Multi Layer Perceptrons (MLP). MLP had the best outcomes in that study, with an efficiency of 79.26% and an F1 score of 76.45%.

In [14] used Adversarial Environment Reinforcement Learning (AERL) as an alternate approach to traditional oversampling techniques. They trained two agents to compete with one another under their proposed paradigm. Defending party and external influences. The attacker agent must forecast attack labels from a certain batch. When the attacking agent is proven to have properly predicted the outcome, it receives one reward point. To increase prediction mistakes, the environment agent must provide the attacking agency with difficult data.

The agent receives a reward point for their efforts. As a result, both agents are going through competitive training. Following training, the attacker agents are then employed to forecast the class labels for the testing dataset. The attacker agent has received adversarial training, allowing it to recognise each specific class with accuracy. Other Ml algorithms with the best oversampling did not perform as well as this AERL model. The obtained F1 score was 0.7940. This strategy was developed upon by a different research team [15], who combined SMOTE and AERL to improve the results. With an F1 score of 0.8243, their results were the best of all the cutting-edge techniques using reinforcement learning models.

The [16] used a Bayesian network and an ANN to create a classification algorithm. They used the NSLKDD and KDD cup 99 databases. They solely considered two-way categorization and concentrated on the testing set (using different partitions). [17] used ensemble classifiers (C4.5, RF, and Forest PA) and the Correlation-based Feature Selection (Bat) method to construct a framework. The KDDTest+ results for the proposed systems were 87.37%. [18] built an adaptive voting system and used a

DNN in conjunction with other basic classifiers on particular features to reach 85.2% accuracy. This is another novel way of classifying votes. In this study, the weight of the voting classifier was determined using 10-fold cross-validation on the training datasets.

Review Aspects of DM Practices

In datasets, association-rule mining reveals surprising relationships between a group of attributes [19]. Associative rules can be used to display the connections between the datasets. Making strategic judgments on a range of actions, such as promotional pricing, shelf management, and other things, can be done using this information [20]. Traditional association rule mining uses datasets from diverse companies to give data analysts the ability to uncover patterns or association rules between datasets [21]. These enormous datasets can successfully be subjected to sophisticated analysis [22], but the security of the data owner [23], whose personal information can be acquired by the dataminer [24], is in jeopardy. Association rule mining is currently one of the most often used methods for detecting patterns in KDD. The standards state that all that is required to solve an ARM problem is to browse the items in a database.

ARM methods can be generically categorised into BFS (Breadth First Search) and DFS (Depth First Search) approaches depending on how the search space is analysed. The BFS and DFS techniques are further separated into Counting and Intersecting on the basis of how support values for the itemsets were arrived at. The Apriori, Apriori-TID, and Apriori-DIC algorithms are based on BFS with Counting strategies, whereas the Partition approach is based on BFS with Intersecting strategies. While ECLAT is based on DFS with Intersecting, the FP-Growth algorithm is based on DFS with Counting methods. The Apriori algorithm is the one that is most frequently employed in BFS with Counting Occurrences among these techniques. Before calculating the number of supports, it eliminates the candidates with infrequent subsets, taking advantage of an itemset's downward closure property. Support and confidence should be considered when analysing the association's rules. By supplying the support values in advance for all subsets of the candidates, BFS provides the necessary optimization.

The disadvantage of this approach is the escalating computing complexity of extracting rules from a sizable database. The Apriori technique has been spread, enhanced, and secured using the Fast Distributed Mining (FDM) algorithm. Organizations may now use data more effectively because to improvements in data mining techniques. By conducting a single database search, the candidates with a cardinality k are enumerated in Apriori. The most important part of the Apriori Algorithm in each transaction is the candidate search. A hashtree structure is employed for this function [25]. Apriori-TID is an extension of Apriori and represents each transaction based on the current candidates it consists of, as opposed to normal Apriori, which uses raw database data. Apriori-Hybrid combines the benefits of Apriori with Apriori-TID. A further Apriori variant called Apriori-DIC aims to obfuscate the distinction between the counting and candidate generating procedures. Prefix trees are used in this process.

DFS with Counting Occurences: For each collection of candidates that has the right size, Counting runs a database search. The simple DFS plus Counting Occurrences combination is practically useless since database scanning involves additional computing overhead. On the other side, FP-Growth makes use of the FP-Tree, a highly compressed representation of transaction data. By calculating the DFS and counting the occurrences, an FP-Tree is produced.

Open Issues and Gaps Identified from Related Works

In this regard, we must address a few inquiries, like:

Should privacy-protecting measures alreadyavailablewith data mining algorithm is really efficient? or ML/DL algorithms are needed?

- Is there any framework that adapts over the interoperability of the system with any of platforms.
- Investigating how they might be applied in real time network applications like Internet of Things.
- Data mining techniques with ML needs to be proved whether they integrate confidentiality, privacy, and trust of the applicable system.

Advancements in data mining and privacy research are required to address these issues. Planning carefully is essential when developing flexible systems. Few applications might call for pure data mining methods, and even fewer would need ML-enhanced data mining that protects user privacy. Therefore, flexible data mining methods and ML/DL algorithms are required to suit changing requirements. The suggested research for DM with ML for network anomaly identification has progressed as a result.

III. PROPOSED METHODOLOGY

Based on Data Mining System: The SO Algorithm

Speculating on A Dos Attack:

The appearance of Dos attack vectors: During the development and production phases of a system, there will be some software and system flaws. Attackers undertake malicious assaults using these defects in an effort to gain an advantage or accomplish a certain objective. Features of Dos assaults Dos attack exhibit the following traits: They are easy to execute, have a wide attack range, a lot of attack intensity, a decent attack concealment, and a great distribution of attack sources. They are also difficult to resist against. Dos attack tools are widely available, making them accessible to anyone with little background in network security. Anyone can download pre-made tools from the launch and Internet attack on specific targets at their discretion without much computer or network experience. Furthermore, offensive data packets are difficult to differentiate from conventional especially storm attack information packets, information packets, which employ several information packets in their attack Denial-of-service attacks are thus extremely difficult to prevent from the standpoint of the "injured party.

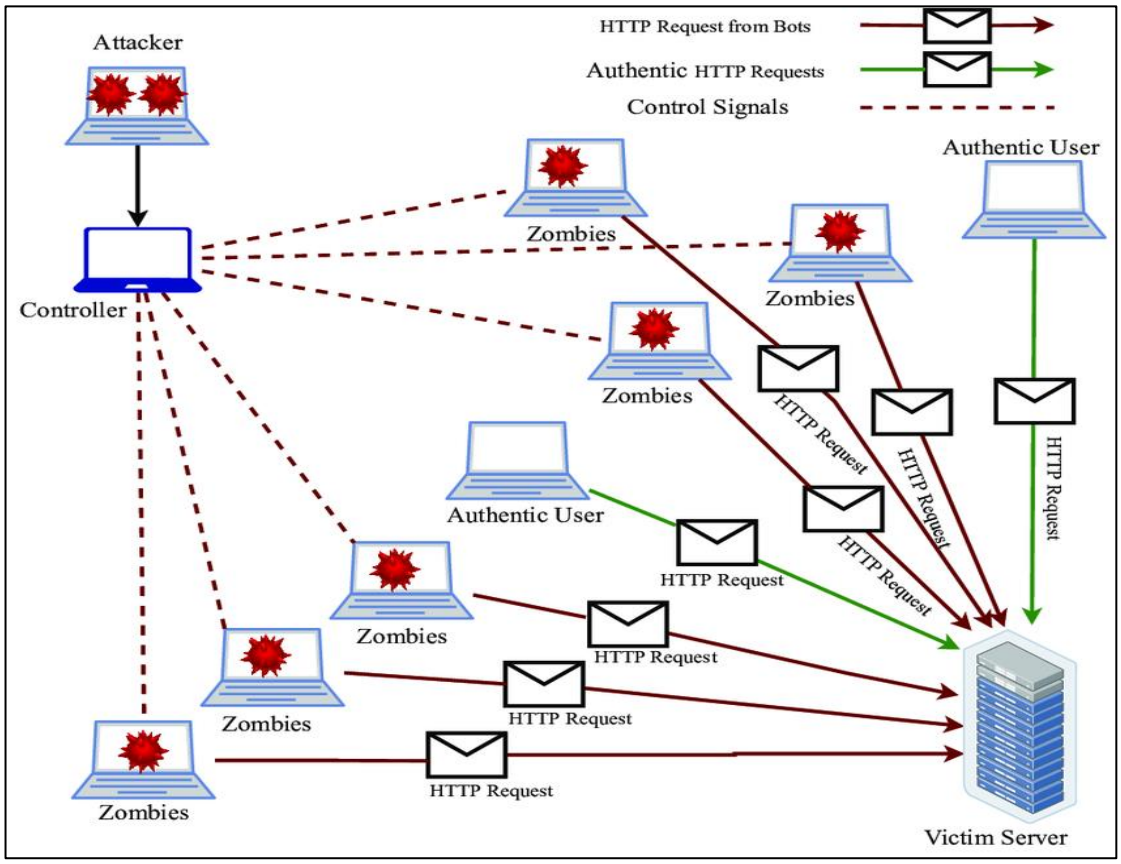


Fig 1. The Structure of Distributed Denial of Service Attack

Different type of Dos assaults can be distinguished: service interruption, bodily harm, and resource exhaustion. Resource exhaustion is a term used to describe an attack in which the attacker uses some technique to completely saturate the target server's service resources, such as its processor, bandwidth, and RAM. When an attacker brings down a service using server-side bugs, it is referred to as a "disrupted service." The phrase "physical break kind" refer to a rejection attack that physically harms goal server by either destroying the routers or network cable directly, by using fire or water to short out or cut off the necessary networks wire, or both. The first type of attack referred to as a virus packet attack, involves sending a sizable volume of abnormal data depending on design faults and vulnerabilities in the target server. Additionally, it causes the wounded person's system to malfunction and crash. The following kind is known as a "storm assault," and it involves flooding the target system with a huge amount of information packets until the goal server is compromised. Storm attacks can occasionally be used in distributed Dos attacks. A "redirected assault," also referred to as the third sort of attack, alters certain network parameters as opposed to deluging the target with virus packets or storms. The DNS server cache, for example, allows data packets sent by or to the injured party to be redirect to different locations.

Dos Tenet attack: In a denial-of-service assault, for instance, the attacker looks for machines that are easy to scan, are not controlled securely, and are unsafe before deleting and changing these puppet machines to blend in. By delivering a sequence of control commands to the puppet computer, the attacker who tries to influence the machines can quickly or at a predefined time send a sizable volume of unwanted content to the target server. Setting a timer for the puppet pc to attack goal computer and launching the attack after that is an alternative. The particulars are shown in **Fig 1.**

Avoiding and identifying Dos attack: It is presently no comprehensive method for preventing and identifying Dos attack; typically, the system is upgraded or patched. Four categories of defence tactics exist. The first step is attack detection, which involves keeping an eye out for Dos attacks, recognizing them, alerting users, and activating an attack response mechanism. The second includes using message filtering to reduce attack traffic after a server has been the target of a Dos attack, speed limitations, etc. to decrease its effects and restore regular server operation. To ensure that the service is available, this function needs to be created as quickly as possible. To identify the last genuine host or the person who started the Dos, employ the third type, often known as a source trace. The fourth technique is known as assault defence, and it contends that defensive measures like patching and encryption must be finished ahead of time in order to decrease the number of attackers in accordance with the formation circumstances and Dos attack principles.

Currently, hostile network layer attacks could stop data packets from travelling over the channel, which is the state of Dos attacks. For instance, a compromised networked router can prevent the sending or receiving of data packets. Packet loss can also result from Dos assaults. Formula (1) can be used to record the Dos assault state if the whole duration is proportionate to the full assault instance's confinement, and it meet the requirement of Formula (2).

$$\{\theta_S(t) \in \{0,1\}\}_{t\in Z_0,} \tag{1}$$

$$Q\left[\sum_{t=0}^{n-1} \theta_S(t) \le n + \frac{n}{\omega}\right] = 1\,, n \in Z \tag{2}$$

At t∈Z0, one of these times, the sensor is about to send the controlled item with the information package. At time t, θ(t)=0 indicates a successful data packet transmission in the channel whereas θ(t)=1 indicates a failed data packet transmission, suggesting that DoS attack is now being launched against the channel. Additionally, since n data packets are sent, Dos attack can only cause (n+ωn) or more data packets to be damaged. Noting that when n=0, θ(t)=0, or both, there isn't a Dos attack at the beginning. Since the attacker's energy constraints prevent it from completely causing all of the data packets to be lost, the average number of data packets that are vulnerable to Dos attacks is 1/ω.

Technology for Data Mining.

One of the most crucial big data technologies for societal growth is information mining technology. More precise and efficient data information may be processed as a result of the growth and application of computer mining technology, producing results that are more thorough. Big data applications can create visual information processing technology by looking at real-world applications. The most crucial feature is its ability to analyze data computationally Data mining technologies presently employ the association algorithm, NN algorithm, regression algorithm, online data mining algorithm, clustering method, and classification algorithm. The SO algorithm, a neural network approach that was the focus of this study, simulates the learning and reasoning processes of the human brain.

We developed a comparable method that can discriminate between various sample data after training on a few samples. Data mining technology extracts information rules from enormous, noisy, and large-scale data that people cannot simply reflect intuitively since all of that information will use future rules. Traditional data mining methods frequently begin with a few straightforward ones. Internet usage is widespread, and methods for analyzing huge data from networks are constantly changing. Today's culture makes extensive use of data analysis tools that combine traditional data mining techniques with Internet infrastructure. By rapidly and accurately locating potentially useful information for users as well as a variety of behavioral information hidden in the network, network data mining technology enables users to access a variety of services suited to their needs.

Finding the purpose of data mining is one of the key objectives of data analysis, and **Fig 2** illustrates computer data mining technology's fundamental procedure. First off, the primary techniques for setting goals include conducting user interviews, investigating and analyzing significant executives in various departments, identifying the overarching business objectives, and conducting studies of business understanding Depending on their level of business experience, data mining engineers must complete administrative, data collection, and "data interpretation" activities. Since checking and editing data, acquiring and organising data using network information, and converting and altering data are all required, data preparation takes significantly

more time. When the update is complete, a data model can be built on top of it. To meet the goals of the data mining cycle, modelling must be optimised numerous times after it is finished. New data mining models and methods are continually being developed, and the number of user apps is increasing. Until corporate limits are put in place, mining technologies probably won't be used frequently.

The SO Algorithm

The Kohonen network serves as the basis for the SO algorithm. Sometimes abbreviated to KN, It was developed by Finnish University of Helsinki Professor Teuvo Kohonen. It is a self-organizing competitive neural network(NN). The ability of the network to learn unsupervised and independently, as well as to change element values through self-organizing function relationships in response to environmental factors, is referred to as self organization. This quality allows the neural network to automatically aggregate and classify data. Neurons will strengthen the impression and match a certain input form in this type of representation, resultant in sensitive. Therefore, through self-organized training and learning, the input form can be divided into a number of clusters. The neurons can work as a detector in a specific environment since each cluster responds to the input form differently.

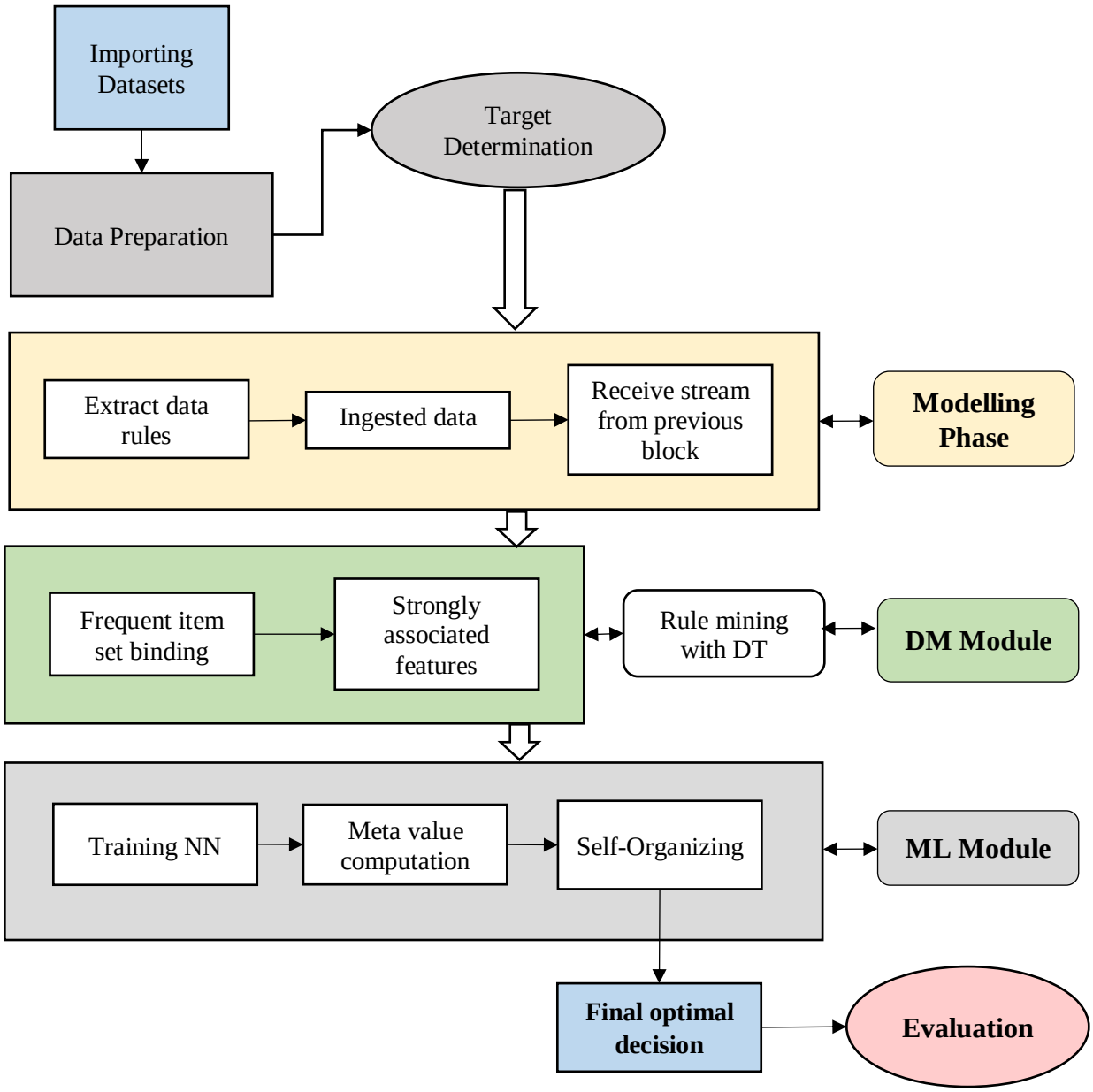


Fig 2. Proposed Data Mining Flowchart

The Euclidean distance, also known as the article's meta value, between the input node of the neurons node of the output layer and the neurons node of the output layer is calculated when data is fed into the neural network.. The SO approach is built around this. Victory cell refers to the neuron with the smallest Euclidean distance. It is possible to change the cell rate coefficient of the surrounding neurons to bring their values closer to those of the input sample. After repeated training, there is a specific link between the element value distributions associated with each neuron.

Similar element value factors are observed in neurons of the same sort, whereas the value coefficients of various types of neurons show substantial variations. This distribution gathers these trends and divides comparable patterns in input values into various groups of neurons. As a result, similar-type neurons start to clump together more frequently. Following practise, the efficiency of the corrected cell values as well as the cell values of neighbouring neurons both continue to decline. The KN algorithm is a popular method for computing using neural network properties.

The classification capabilities of the conventional unsupervised KN algorithm are constrained. Unsupervised classification of the same piece of data using ill-defined categories is possible, but the classification results will correlate to different network nodes. For one-to-one linked node categories, the KN classification categories are greater than the actual data categories. A supervised learning phase follows the initial phase of unsupervised classification in the modified SO technique presented in this research, enabling the system to adjust to the exact classification findings and enhancing the classification impact. The primary concept of the approach is shown in **Fig 3.**

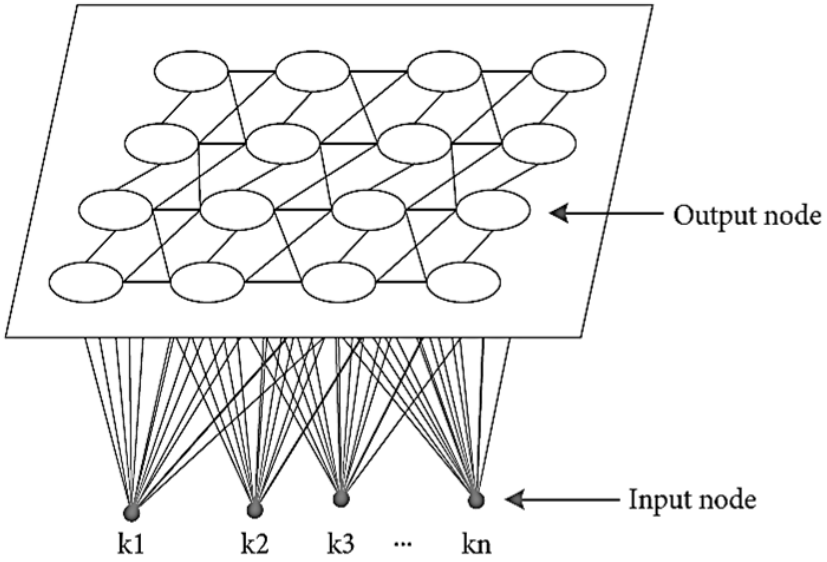


Fig 3. The Diagrammatic Representation AKN Algorithms

According to **Fig 3**, The node is designated as k, and if the input node is at the bottom, computation has occurred. S KN output nodes are denoted by the symbol f, and there are also output nodes, which are denoted by the symbol u. Full nodes are present if the input vector contains more than n components. The component value size (abbreviated as CKf) from the input node to the output node can be used to connect the output nodes on the same plane to one another. The SO technique, which uses a huge amount of samples for free wisdom and endlessly updates component rate of network to finally decide the distribution of data clustering, speeds up the supervised learning process.

Content of SO algorithm: Steps of the algorithm are as below: Input: Samples used for training and testing; Output: accuracy of the test sample, sample component value coefficient matrix after training, clustering category. A random number is used to value the communication component between the input and output nodes, combined with expression of neighborhood $A_{f,f(x)}(s)$and training rate $\eta(s)$sum, S is the amount of output neurons. t0 is the beginning, t1 is a fixed value, indicating time and the total amount of training is t2.

Input mode of Network

$$Z = [z_1, z_2, \dots \dots, z_K]^{\mathrm{T}} \tag{3}$$

Among them, K is the input vector's dimension.

Setting the weight to zero:

$$C_f = \left[C_{1f}, C_{2f} \ldots, C_{kf}\right]^T \tag{4}$$

Among them, f = 1, 2, . . ., S.

Computing the element value and sample vector

$$d_f = \left|Z - C_f\right| = \sqrt{\sum_{u=1}^{5} \left(x_u - C_{uf}(t)\right)^2} \tag{5}$$

Then, The victory element f^*'s element value expression is

$$d_{f=} = \mathrm{m}\ \left\{d_f\right\} \tag{6}$$

Component values vectors of nodes that are connected in the outputs and geometrical neighborhood is adjusted:

$$C_f\,(n+1) = \eta(s) A_{f,f(n)}(n)\left(Z - C_f(n)\right) + C_f\,(n) \tag{7}$$

Among them, The neighborhood adjusting function is denoted by $A_{f,f(x)}(s)$.

Return to step (2) if there remains a training sample dataset;

Return to step (2) if all sample are trained and $t < t2, t = t + 1$; else, back to step (8).training is over.$\eta(s)$ is the train value at s, as stated in (8), and The value of s grows along with the number of trains. As a result, the inversely proportional to the relationship predicted by (8) and will continue to decline.

$$\eta(s) = \eta_1 \mathrm{ex}\left(1 - \frac{s}{t_2}\right) \tag{8}$$

The overall number of training among them is t2, and the beginning training value is $\eta_1 \cdot s =$ 0,1,2..Neighborhood's redevelopment has carried out by,

$$A_{f,f(x)}(s) = \mathrm{ex}\left(1 - \frac{d_{fu}^2}{2\varepsilon(s)^2}\right) \tag{9}$$

d_{f,w^*} and $\varepsilon(s)$.stand for the parameter controlling the data distribution and the distance between each neuron on same plane and the winning element, respectively. The formula for correction is σ.

$$\varepsilon(s) = \varepsilon_1 \mathrm{ex}\left(1 - \frac{n}{t_1}\right) \tag{10}$$

The formulas (11)–(13) show how training and testing data are normalised. It serves as an example of how training and testing data has been normalised.

$$x = \frac{(x_{\max} - x_{\min}) \times (y - y_{\min})}{y_{\max} - y_{\min}} + x_{\min} \tag{11}$$

$$x'_{kf} = \frac{x_{uf}}{\sqrt{x_{1f}^2 + x_{2f}^2 + \cdots + x_{kf}^2}} \tag{12}$$

$$C'_{kf} = \frac{C_{uf}}{\sqrt{C_{1f}^2 + C_{2f}^2 + \cdots + C_{kf}^2}} \tag{13}$$

The degree of cluster between the output of neurons and the input model is represent by the following formula (5), where the Euclidean distance d_f equals

$$\begin{aligned} \rho\left(C_f(Z)\right) &= \left\{\rho\left(d_f(Z)\right), 0, \sqrt{0.2}\right\}, \\ \rho\left(C_f(Z)\right) &= e^{-d_f(Z)^2/0.4}. \end{aligned} \tag{14}$$

When $\rho > 0.5$, The output neurons is the cluster to which the input model belongs. According to the group, each group's mean price is determined as shown in the following equations:

$$\bar{d}_x = \frac{x'_{kf}}{x_{1f} + x_{2f} + \cdots + x_{kf}} \tag{15}$$

(16)

$$\bar{d}_C = \frac{C'_{kf}}{C_{1f} + C_{2f} + \cdots + C_{kf}}.$$

The neuron centre score of the cluster can be obtained using formula (17) and the cluster information computation model .

$$d(f,u) = \sqrt{C_1|x_{1u} - x_{1f}|^2 + C_2\,|x_{2u} - x_{2f}|^2 + C_3|x_{3u} - x_{3f}|^2 \ldots . + C_k\,|x_{ku} - x_{kf}|^2}. \quad (17)$$

IV. NUMERICAL RESULTS AND INTERPRETATIONS

The N-BaInternet of Things dataset [24], one of the most recent datasets created in 2020 and made available is the nbaInternet of Things dataset (https://www.kaggle.com/mkashifn/dataset), was utilised in the research to identify Internet of Things botnet attacks. Wireshark's main button was used to record nine Internet of Things sensor connections on a local network for the dataset. Its 115 statistically derived properties were from pcap records and were acquired. Each of the seven statistical variables was calculated over five different time periods (count, magnitude, radius, mean, variance, correlation coefficient and covariance). This dataset features a time frame, making it suitable for Intrusion Detection System (IDS). Information is obtained from traffic over a predetermined time period over the similar time phase. But for each of these four attributes, three or more statistical measures have been developed, resulting in a total of twenty-three features. A total of 229,899 samples were used, of which 13,133 were employed in harmful attacks and 216,236 were employed in lawful ones. Following are some samples from the malicious attack: 27,188, 9502, 23,361, 15,148, 26,210, 21,205, 24,250, 21,995, 23,755, 24,102, and Gafgyt.udp, Gafgyt.tcp, Gafgyt.combo, Gafgyt.junk and Gafgyt.scan. In this study, a multi-class dataset with a total of 11 classes was used.

Performance Evaluation

The preliminary test results served as the foundation for the experimental results. The dataset is divided using the hold out approach. In this case, 25% of the data are used for testing while 75% are used for training. In Sections 3.1 and 3.2, a table with confusion matrices and performance comparisons of classification algorithms is displayed. Evaluations of various Train-Test techniques, It contain the Receiver Operating Characteristic (ROC) Curve as well as comparisons with previous studies. Once characteristics are selected with RF and categorized with proposed, some statistical tests are conducted to assess the efficacy of the suggested technique. (I) RF-RF (RF categorization based on the selection of RF signals), (II) RF-RFE (RF classification utilising feature selection based on RFE), (III) RF-RFECV (RFECV-based feature selection for RF classification), (IV) RF-SelectK (choosing the K-best features for RF classification) and (V) RF-WFS (When classifying RF, use WFS) are the other five ML schemes that have been used (No feature selection in RF classification). **Table 1** displays the categorization performance.

Table 1. Evaluation of Performance of Various Techniques

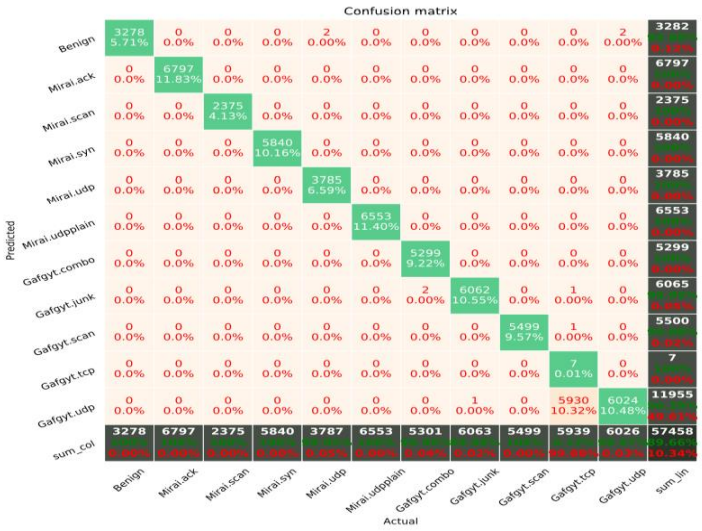
ML Schemes	ACC	Kappa	F1_Score	Kappa	MCC	Balanced Accuracy	Threat_Score	Sensitivity	Specificity
RF-RF	89.6%	88.5%	86.2%	88.5%	89.6%	86.3%	98.9%	89.6%	89.6%
Proposed	**99.9%**	**99.9%**	**99.9%**	**99.9%**	**99.9%**	**99.8%**	**99.9%**	**99.9%**	**99.9%**
RF-RFE	89.6%	88.5%	86.2%	88.5%	89.6%	86.3%	98.9%	89.6%	89.6%
RF-RFECV	89.6%	88.5%	86.2%	88.5%	89.6%	86.3%	98.9%	89.6%	89.6%
RF-Selectk	89.6%	88.5%	86.2%	88.5%	89.6%	86.3%	98.9%	89.6%	89.6%
RF-WFS	89.6%	88.5%	86.8%	88.5%	89.4%	86.6%	98.9%	89.4%	89.6%

Table 1 shows that the accurateness of the proposed model was 99.9%, while that of the models RF-SelectKBest, RF-RF, RF-RFE, RF-WFS and RF-RFECV was approximately 91%. Other performance indicators, includes sensitivity (99.9%), specificity (99.9%), F1-score (99.9%), evenhanded accuracy (99.9%), etc., have demonstrated that proposed also produces the greatest outcomes. With regard to error rating, the proposed model is the least flawed (0.07 percent). It may be concluded that the recommended model has outperformed the other models by a wide margin. But out of all of these models, the RF WFS method did the poorest. We have also estimated how long it will take to put our suggested strategy into practice. The entire test

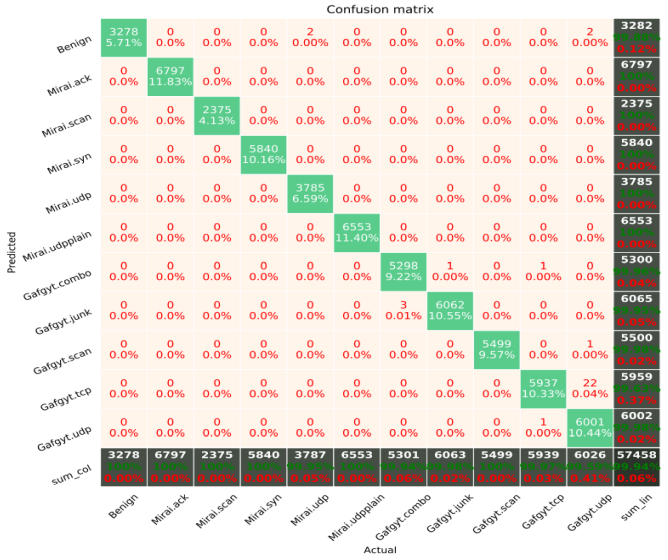
set was completed in 57.823 seconds on a machine with a Core i9 processor and 64 GB of RAM (57,453 instances). This means that PROPOSED would complete each attack detection in less than 0.0010064 seconds.

Confusion Matrix

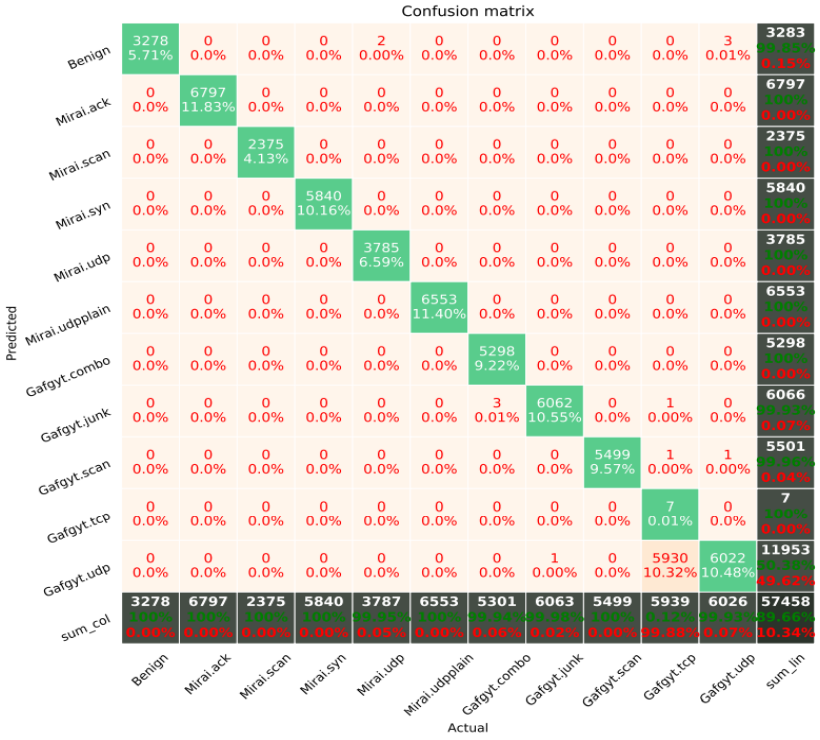
To evaluate how well various classifiers work, utilise a multi-class confusion matrix. Our N-BaInternet of Things dataset contains a substantial number of classes, so we can use the multi-class confusion matrix to illustrate the confusion that occurs when assaults are expected. Figure 4 displays the confusion matrix for a number of classifiers. There is a justification to utilize the same test set in each of the circumstances depicted in Figure 4. Only 7 out of 5939 Gafgyt.tcp attacks are classified by the classifier, which is exceptional according to the confusion matrix of [i] RF-RF. Once again, the majority of Gafgyt.udp attacks are categorized using RF-RF. The accuracy rating is lowered because 5930 is mistakenly identified as Gafgyt.tcp Our proposed method [ii] Suggested addresses the unexpected concerns brought up by the aforementioned classifier to distinguish between attacks on Gafgyt.udp and Gafgyt.tcp respectively. This model's PROPOSED has an extremely low misclassification rate. 99.9426% of assaults are accurately identified by the model. It is abundantly clear that Gafgyt.tcp assaults are rarely classified by the [iii] RF-RFECV confusion matrix. Here, it incorrectly classifies 7 out of 5939 attacks as Gafgyt.tcp while incorrectly identifying 5930 assaults as Gafgyt.upd. The method we advise is substantially more accurate than RF-RFECV, which has an accuracy of 89.6585%. Furthermore, the RF-WFS assaults [iv], [v], and [vi] frequently misclassify.



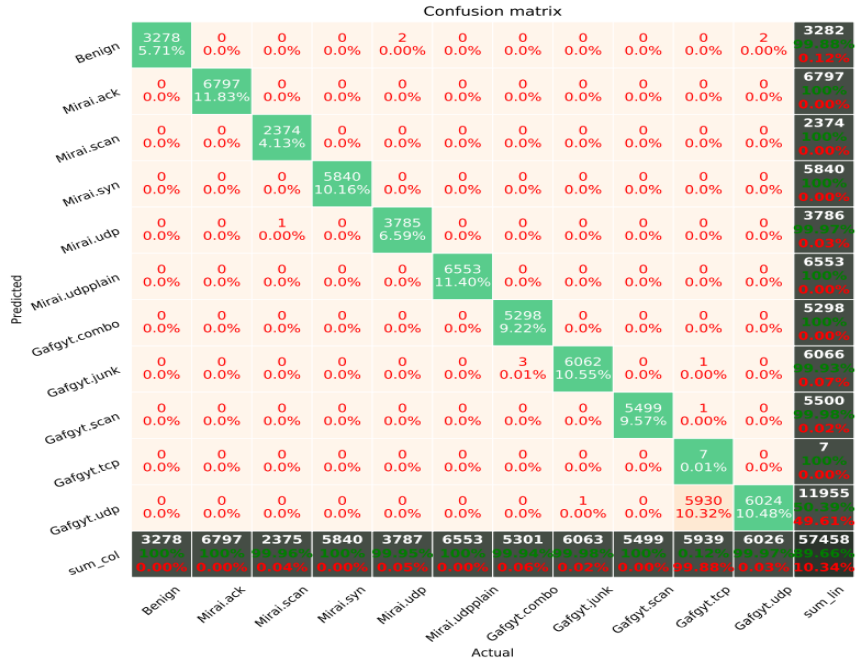
[i]



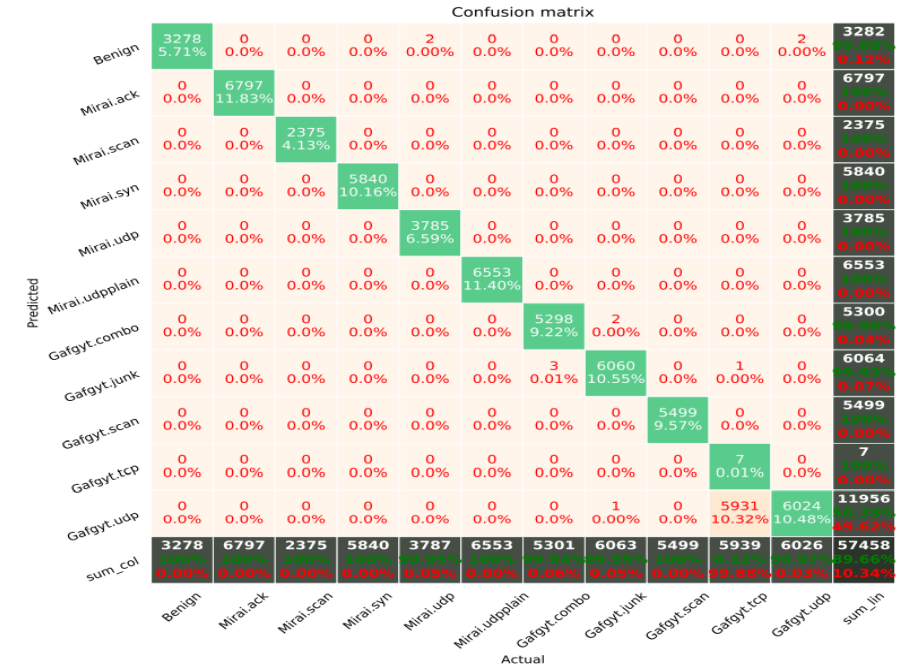
[ii]


Confusion matrix
Predicted
Actual
Benign
Mirai.ack
Mirai.scan
Mirai.syn
Mirai.udp
Mirai.udpplain
Gafgyt.combo
Gafgyt.junk
Gafgyt.scan
Gafgyt.tcp
Gafgyt.udp
sum_col
sum_lin

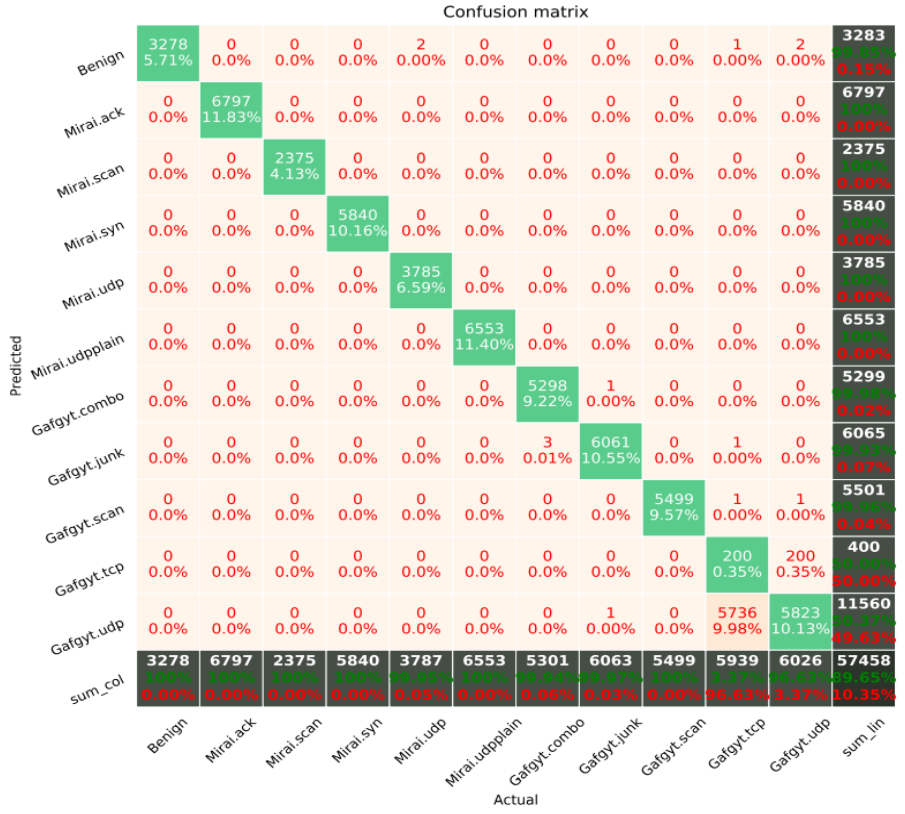
[iii]


Confusion matrix
Predicted
Actual
Benign
Mirai.ack
Mirai.scan
Mirai.syn
Mirai.udp
Mirai.udpplain
Gafgyt.combo
Gafgyt.junk
Gafgyt.scan
Gafgyt.tcp
Gafgyt.udp
sum_col
sum_lin

[iv]



[v]



[vi]

Fig 4. Confusion matrix of: [i] RF-RFECV, [ii] RF-SelectKBest, [iii] RF-WFS, [iv] RF-RF, [v] RF-RFE and [vi] proposed. The Y-axis represent the classifier's anticipated tag, while the X-axis represents the label.

Evaluation of Various Train-Testing Techniques

As was already mentioned, the suggested strategy splits the data into training and testing, using 75% of the data for each. According to our prior experience, data were primarily classified as 75-25%. We also assess the performance when utilising the 70-30% and 67-33% train-test splitting strategies. The outcome shown in **Table 2** demonstrates a negligibly small impact on data splitting.

Table 2. Various Train-Testing Techniques

Performance	Training-Test (67–33%)	Training-Test (70–30%)
F1_Score	99.9%	99.9%
Accuracy	99.9%	99.9%
Threat_Score	99.9%	99.9%
Specificity	99.9%	99.9%
Kappa	99.9%	99.9%
Balanced Accuracy	99.9%	99.9%
Sensitivity	99.9%	99.9%
MCC	99.9%	99.9%

ROC Curve

Fig 4 shows that all of the classes, including Gafgyt.udp class, were correctly classified using our hybrid proposal. The Gafgyt.udp class is not correctly categorized by further approaches including RF-RFECV, RF-RF, RF-SelectKBest, RF-RFE and RF-WFS. Figure 10 compares the Gafgyt.udp receiver operating characteristic curve (ROC) to all other classes. It is obvious that the proposed method outperforms earlier methods at detecting Gafgyt.udp. Due to their similar performance, RF-RFE, RF-RFECV, RF-SelectKBest, the ROCs for RF-WFS and RF-RF overlap. In addition, how AUROC is relevant to statistical p values and comparable to statistical Mann-Whitney U-statistic testing because, as we noted in our paper, the ROC of our suggested technique is roughly 1, which is a perfect classifier, and as a result, we can say $p<0.001$. We did not perform an ANOVA test or other statistical analysis to establish whether they are statistically significant.

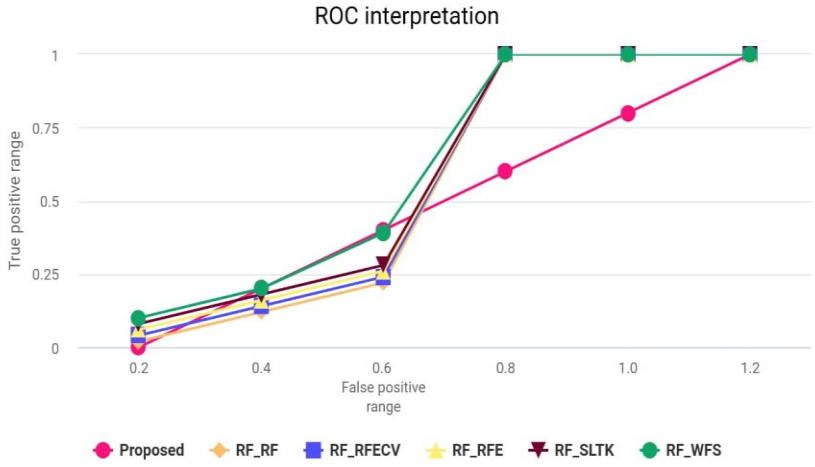


Fig 10. Gafgyt.udp's ROC curve in comparison to all classes Take note of the overlapped RF-RFECV, RF-WFS, RF-SelectKBest, ROCs for RF-RF and RF-RFE.

Performance Evaluation in Relation to Other Studies

Five of the alternative methodologies employed the same N-BaInternet of Things dataset, and On various performance indicators, we contrast the performance of our suggested model with that of earlier research. Therefore, we used those studies to compare. Adeel Serpil also employed RF and DMLP on the same CICIDS2017 dataset and achieved accuracy rates of 99.67% and 91%, respectively. Kathleen employed SVM-Decision Tree-NB (SVM-DT-NB) classifiers once more

on the KDDCup99 dataset, and the accuracy was 99.62%. On the N-BaInternet of Things dataset, Yan, Chaw, Abdulkareem, and Tran also used RNN, Collective Deep Learning, Classification and Regression Trees, and Local-Global Best Bat Algorithm for Neural Networks (LGBA-NN) approaches, yielding accuracy values of 99.10%, 99%, 89.75%, 99.84%, and 90%, respectively. Abdullah, on the other hand, did not use any of these approaches. Modern classifiers were used in the majority of the aforementioned approaches however their performance was still inferior to that of our suggested method. A hybrid ML model is proposed that selects the crucial characteristic division and improves classification accuracy. This is because the suggested changes eliminate superfluous and pointless functionality. As a result, it offers an improved decision border, improving classification accuracy and reducing runtime. **Table 3** demonstrates that in this situation, our proposed model outperforms every previous approach.

Table 3. Evaluation in Relation to Other Studies

Classifiers	Accuracy	Dataset
SVM-DT-NB	99.6%	KDDCup99
RF	99.6%	CICIDS2017
NB-J48-ANN	99.1%	N-Ba Internet of Things
Collective Deep Learning	99.8%	N-Ba Internet of Things
LGBA-NN	90.2%	N-Ba Internet of Things
RNN	89.7%	N-Ba Internet of Things
DMLP	91.1%	CICIDS2017
CART	99.1%	N-Ba Internet of Things
Proposed	99.9%	N-Ba Internet of Things

V. CONCLUSION AND FUTURE WORK

The paper sets up the SO algorithm model, examines Dos attack data samples, identifies patterns in the data, contrasts the data with the conventional KN algorithm, and concludes that the AKN algorithm-based computer data mining technology is employed in the recognition and defence against DoS assaults. The XGB-accuracy, RF's sensitivity, and Kappa index are a few instances of measurements that have all been found to be almost 10% more accurate than those of other systems. Effectiveness was assessed with greater than 99% accuracy on the N-BaInternet of Things dataset, outperforming other cutting-edge machine learning techniques. This suggested method can significantly enhance the security components of Internet of Things systems because the security and confidentiality of Internet of Things devices are essential to their success. However, 383,379 finding previously unknown hazards has proven to be difficult because new assault kinds are always developing. Our present methodology takes 0.0010063 seconds to identify a single attack. Future study might concentrate on enhancing accuracy while speeding up detection times to permit application in a busy Internet of Things system. Additionally, we'll assess how well machine learning classifiers perform in locating unreported assaults in Internet of Things environments. As a result, the SO algorithm-based computer data mining technology can offer Dos attack defence detection that is quicker, more effective, and more exact.

Data Availability

The Data used to support the findings of this study will be shared upon request.

Conflicts of Interests

The author(s) declare(s) that they have no conflicts of interest.

Funding

No funding was received to assist with the preparation of this manuscript.

Ethics Approval and Consent to Participate

The research has consent for Ethical Approval and Consent to participate.

Competing Interests

There are no competing interests.

Reference

[1]. Bhuyan, M.H., Bhattacharyya, D.K., Kalita, J.K.: Towards generating reallife datasets for network intrusion detection. IJ Network Security. 17(6), 683–701 (2015)

[2]. J. Jang-Jaccard and S. Nepal, "A survey of emerging threats in cybersecurity," Journal of Computer and System Sciences, vol. 80, no. 5, pp. 973–993, Aug. 2014, doi: 10.1016/j.jcss.2014.02.005.

[3]. Uppal, H.A.M., Javed, M., Arshad, M.: An overview of intrusion detection system (IDS) along with its commonly used techniques and classifications. Int J Comput Sci Telecommun. 5(2), 20–24 (2014)

[4]. N. Sun, J. Zhang, P. Rimba, S. Gao, L. Y. Zhang, and Y. Xiang, "Data-Driven Cybersecurity Incident Prediction: A Survey," IEEE Communications Surveys & Tutorials, vol. 21, no. 2, pp. 1744–1772, 2019, doi: 10.1109/comst.2018.2885561.

[5]. P. Mishra, V. Varadharajan, U. Tupakula, and E. S. Pilli, "A Detailed Investigation and Analysis of Using Machine Learning Techniques for Intrusion Detection," IEEE Communications Surveys & Tutorials, vol. 21, no. 1, pp. 686–728, 2019, doi: 10.1109/comst.2018.2847722.

[6]. M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Jul. 2009, doi: 10.1109/cisda.2009.5356528.

[7]. R. and P. P., "Deep Learning With Conceptual View in Meta Data for Content Categorization," Advances in Computational Intelligence and Robotics, pp. 176–191, 2021, doi: 10.4018/978-1-7998-2108-3.ch007.

[8]. Yin, Y. Zhu, J. Fei, and X. He, "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks," IEEE Access, vol. 5, pp. 21954–21961, 2017, doi: 10.1109/access.2017.2762418.

[9]. Z. Li, Z. Qin, K. Huang, X. Yang, and S. Ye, "Intrusion Detection Using Convolutional Neural Networks for Representation Learning," Lecture Notes in Computer Science, pp. 858–866, 2017, doi: 10.1007/978-3-319-70139-4_87.

[10]. J. Yogapriya, C. Saravanabhavan, R. Asokan, Ila. Vennila, P. Preethi, and B. Nithya, "A Study of Image Retrieval System Based on Feature Extraction, Selection, Classification and Similarity Measurements," Journal of Medical Imaging and Health Informatics, vol. 8, no. 3, pp. 479–484, Mar. 2018, doi: 10.1166/jmihi.2018.2326.

[11]. Z. Chen, C. K. Yeo, B. S. Lee, and C. T. Lau, "Autoencoder-based network anomaly detection," 2018 Wireless Telecommunications Symposium (WTS), Apr. 2018, doi: 10.1109/wts.2018.8363930.

[12]. M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Conditional Variational Autoencoder for Prediction and Feature Recovery Applied to Intrusion Detection in IoT," Sensors, vol. 17, no. 9, p. 1967, Aug. 2017, doi: 10.3390/s17091967.

[13]. Preethi. P and Asokan. R, "Neural Network Oriented RONI Prediction for Embedding Process with Hex Code Encryption in DICOM Images," 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Dec. 2020, doi: 10.1109/icacccn51052.2020.9362880.

[14]. G. Caminero, M. Lopez-Martin, and B. Carro, "Adversarial environment reinforcement learning algorithm for intrusion detection," Computer Networks, vol. 159, pp. 96–109, Aug. 2019, doi: 10.1016/j.comnet.2019.05.013.

[15]. P. Palanisamy, A. Padmanabhan, A. Ramasamy, and S. Subramaniam, "Remote Patient Activity Monitoring System by Integrating IoT Sensors and Artificial Intelligence Techniques," Sensors, vol. 23, no. 13, p. 5869, Jun. 2023, doi: 10.3390/s23135869.

[16]. KumarShrivas and A. Kumar Dewangan, "An Ensemble Model for Classification of Attacks with Feature Selection based on KDD99 and NSL-KDD Data Set," International Journal of Computer Applications, vol. 99, no. 15, pp. 8–13, Aug. 2014, doi: 10.5120/17447-5392.

[17]. Y. Zhou, G. Cheng, S. Jiang, and M. Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," Computer Networks, vol. 174, p. 107247, Jun. 2020, doi: 10.1016/j.comnet.2020.107247.

[18]. P. Preethi and R. Asokan, "An Attempt to Design Improved and Fool Proof Safe Distribution of Personal Healthcare Records for Cloud Computing," Mobile Networks and Applications, vol. 24, no. 6, pp. 1755–1762, Oct. 2019, doi: 10.1007/s11036-019-01379-4.

[19]. P. D. Shenoy, K. G. Srinivasa, K. R. Venugopal, and L. M. Patnaik, "Dynamic Association Rule Mining using Genetic Algorithms," Intelligent Data Analysis, vol. 9, no. 5, pp. 439–453, Nov. 2005, doi: 10.3233/ida-2005-9503.

[20]. P. Deepa Shenoy, K. G. Srinivasa, K. R. Venugopal, and L. M. Patnaik, "Evolutionary Approach for Mining Association Rules on Dynamic Databases," Lecture Notes in Computer Science, pp. 325–336, 2003, doi: 10.1007/3-540-36175-8_32.

[21]. S. J. Rizvi and J. R. Haritsa, "Maintaining Data Privacy in Association Rule Mining," VLDB '02: Proceedings of the 28th International Conference on Very Large Databases, pp. 682–693, 2002, doi: 10.1016/b978-155860869-6/50066-4.

[22]. S. M. Darwish, M. M. Madbouly, and M. A. El-Hakeem, "A Database Sanitizing Algorithm for Hiding Sensitive Multi-Level Association Rule Mining," International Journal of Computer and Communication Engineering, vol. 3, no. 4, pp. 285–293, 2014, doi: 10.7763/ijcce.2014.v3.337.

[23]. J. Vaidya and C. Clifton, "Privacy preserving association rule mining in vertically partitioned data," Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, Jul. 2002, doi: 10.1145/775047.775142.

[24]. J. Vaidya and C. Clifton, "Secure set intersection cardinality with application to association rule mining," Journal of Computer Security, vol. 13, no. 4, pp. 593–622, Oct. 2005, doi: 10.3233/jcs-2005-13401.

[25]. M. R. B. Diwate and A. Sahu, "Efficient Data Mining in SAMS through Association Rule," International Journal of Electronics Communication and Computer Engineering, vol. 5, no. 3, pp. 593–597, 2014.