Journal of Intelligent Systems and Internet of Things
Vol. 14, No. 01, PP. 102-113, 2025
DOI: https://doi.org/10.54216/JISIoT.140108

Modelling a Constructive Approach For Predicting Attacks Over IoT Network Environment

B. Sowmya[1,*] , Nagendra Muthuluru[1]

[1]Department of Computer Science and Technology, Sri Krishnadevaraya University, Anantapuramu, India

Emails: bsowmya2008@gmail.com; nagendramuthuluru@gmail.com

Abstract

Internet of Things (IoT) devices are more attractive towards various vulnerable activities and nodes are easily compromised by attackers. The complexity of insecure IoT node installation relies on device heterogeneity and resource constraints because of the network ends and conventional endpoints. This work concentrates on modeling an efficient IoT-based preservation model (iPRES) which is a lightweight approach used for detecting anomaly and performance various analyses at the endpoints. This work integrates linear Support Vector Machine for pattern analysis and adaptive fuzzy rule model for data pattern rule generation to examine malicious network functionality and network traffic. While adopting the rules, the compromised node needs to fulfill the generated rules; when it fails then it is considered as malicious activity. Then, the iPRESmodels impose network access restrictions on the compromised and terminate the further process. Thus, the nodes are prevented from further network attacks. The evaluation model is done with the use of an online available network dataset and the dataset samples are evaluated in the complex network scenario. The simulation is done in MATLAB 2020a simulation environment and the accuracy attained with this model is higher compared to other approaches. Similarly, other metrics like False Alarm Rate (FAR) are evaluated for predicting malicious network functionality. The significance of the model is evaluated based on the prediction and mitigation of various network attacks. The anticipated model shows a prediction rate of 90.21% for DoS attacks, 89.13% for R2L, 91.65% for probe, and 93.56% for U2R attacks.

Received: February 12, 2024 Revised: April 22, 2024 Accepted: July 09, 2024

Keywords: Internet of Things; Malicious activities; Security; Attack, Fuzzy rule model; Pattern analysis

1. Introduction

The Internet of Things (IoT) principle is defined as the combination of universally recognizable heterogeneous objects around animals, humans, sensors, cameras, vehicles, and so on by transmitting data devoid of the need for H2H or H2C connections [1]. IoT applications can vary from an essential device for an intelligent household to a complicated apparatus. However, despite their disparate goals, various IoT applications share a standard set of properties [2]. A top node in the Internet of Things can perform data acquisition, transmission, processing, and utilization [3]. Lightweight, less memory, and less-power-consumption sensors with external communications characteristics are used to obtain information about the external surroundings during data acquisition. To connect devices and users along greater distances for data transmission, WiFi, Ethernet, ZigBee, and wire-based technologies are integrated with TCP/IP [4]. Applications process the data to get usable data during the data processing and utilization step. After decision-making for data acquisition, control instructions are executed to influence external surroundings. The risk to E2E privacy in IoT applications is enhanced by integrating numerous technologies, heterogeneity, and the unique communication facilities specified for IoT [5]. Even though multiple approaches enhance data privacy, authentication, and access, IoT devices are vulnerable to various threats which impact the entire network. Intrusion prevention systems and signature-based anti-malware approaches are ineffective against updated and entirely new threats and risks. A protective mechanism that can identify new and possible intrusions is achieved by intrusion detection systems (IDSs) based on anomaly detection [6]. The identification of anomalies does not necessitate the discovery of attack signatures beforehand. Since creating IDSs, a substantial issue for data security, investigators have classified IoT networks into the following categories [7]. IoT network nodes containing IDS agents have less memory, low processing power, and low battery energy capacity than conventional networks whose system administrator employs IDS agents in network elements with significant computational resources capabilities [8].

- In conventional networks, the end systems are linked to routers, switches, and wireless access points directly to transmit packets towards the destination. In IoT, there are various hops, and a primary node can serve as an end system and transfer packets at the same time. Furthermore, the network topology changes regularly (e.g., mobile sinks, VANETs, dynamic CH selection). The topology's uniqueness presents new issues for IDSs.
- IEEE 802.15.4, 6LoWPAN, IPv6 RPL, and CoAP are some of the protocols utilized in IoT networks. Variability in protocols brings additional vulnerabilities, creating new problems for IDSs.
- Because of the characteristics mentioned above of IoT networks, the IDS design must be lightweight. It should be able to run its operations using the resources available in the network's sensor node, and it must be efficient to protect the network from possible threats.

Saraswathi [8] described a lightweight IDS as a tiny, robust, and versatile permanent part of the network security architecture. Ma et al. [9] defined a lightweight system attempt to save energy and minimize computational resources. Yin et al. [10] described a system as lightweight if it completes its operation using low energy and less computational resources. A tiny IDS is developed by excluding difficult feature extraction and feature selection stages to attain the features mentioned above. The research suggests that an intrusion in an IoT network can be identified accurately by this method. Instead of using complex statistical approaches, it is necessary for developing and study such understandable and straightforward algorithms for these applications. Numerous datasets provide samples of DoS attacks in various contexts, such as KDD'99, CAIDA, DARPA, DDoS, etc. [11]. The PDR per node is the network traffic characteristic employed in our approach. However, none of the datasets listed above includes this characteristic. The dataset KDD'99, for example, represents data samples in variables like connection time, protocol type, Land, and so on, however not packet arrival rate [12] – [15]. So this approach cannot be evaluated by using these datasets. The innovation of this paper is in the lightweight IDS design for IoT networks, i.e., lowering the system's cost in terms of energy usage and processing resources. Furthermore, the difficulty of an SVM classifier is directly proportional to the input vector dimensions. If dimensions in the input vector are high, then the problem of SVM becomes high. So, the measurements are minimized by selecting only 2 to 3 features from the available vector. In summary, a lightweight IDS is created using the packet transmission rate from which the features are selected. Implicitly, the energy and time taken by these processes are low when evaluated to a system that evaluates complicated attributes like protocol service, type, land, and incorrect fragments, as described in the NSLKDD dataset. As a result of this approach, the suggested IDS is acceptable in IoT sensor nodes while maintaining system efficiency. The work is structured as: section 2 depicts the extensive analysis of the various existing approaches; section 3 explains the proposed methodology. In section 4, the numerical results and an elaborate discussion of the attained results are given. Section 5 summarizes the work with the idea of the future research extension.

2. Related works

Similar researches done in the IoT field is discussed below. Zhu et al. [16] designed a detector and firewall for IoT by employing integrated approaches such as K-Means and BIRCH [17] for various microservices. Multiple clusters were merged if the center is three times the standard deviation. By using this approach, the accuracy attained is 96.3%. Sahu et al. [18], an intelligent home system identified security vulnerabilities using the DL technique DRNN was employed. The DoS and DDoS attacks architecture are described. Khan et al. [19] designed a detector to identify ON/OFF threats by the malicious node in an IoT framework. The IoT network usually functioned when it was attacked in its inactive or OFF state. Therefore, a light probe routing technique was employed to compute every neighboring node's reliability to recognize an anomaly. Buczak et al. [20] developed a fog-to-things architecture to identify the threat. In this, an open-source dataset performed comparison research of deep NN. The objective was to determine the threat and anomaly of four classes. By the deep neural network approach, the accuracy for threat detection in four categories was 98.27%, and by the external neural network, the process was 96.75%. Tang et al. [21] discussed the safety issues while implementing embedded techniques in IoT. In addition, securing the information transmitted among logical, physical, and virtual elements was difficult. To solve these issues, digital watermarks are employed in this research. Tavallaea et al. [22] developed an intruder identification technique for IoT, in which multiple ML classifiers are employed for detecting network vulnerability scanning and Denial of Service (DoS) threats. By utilizing the software Wireshark, network traffic is monitored for four days to create the data set. Additionally, Weka was employed to apply machine learning classifiers.

Bhattacharjee et al. [23] described the identification of abnormalities in healthcare analytics by utilizing medical image analysis, IoT sensors, extensive data mining, biomedical signal analysis, and predictive analytics. An approach for identifying heart anomalies using a smartphone was also described in this work. Martens et al. [24] developed an approach for intrusion recognition using a two-tier classification module and a two-layer dimension reduction which was achieved by applying PCA and LDA. Here, U2R and R2L attacks were also discovered using this methodology. The entire experiment was conducted with the NSL-KDD dataset, and malicious activity was identified using 2T classification modules, NB and Certainty Factor version of K-NN. Kim et al. [25] employed Uncertainty-managing Batch Relevance-based Artificial Intelligence (U-BRAIN) on the NSLKDD dataset. The model runs on several machines managing missing information. There are 41 features included in the NSLKDD dataset, from which six elements were chosen by the J48 classification approach. The accuracy of NSL-KDD was 94.1%, and of Real Traffic Data was 97.4%. Paul et al. [26] designed a classification-based attack identification process with the cloud. An ELM is applied on the artificial Netflow formatted information created by the IoT network. The operations such as scanning, commanding, and controlling the impacted network were carried out in IoT systems, and their accuracy values were 0.99, 0.76, and 0.95, respectively. Table 1 depicts the comparison of attack prediction methods and samples.

Table 1: Comparison of attack prediction methods and samples [27] – [30]

S. No	Attack	Definition	Samples
1	Distributed DoS (DDoS)	DDoS are attacks and simple to initiate but hard to trace sources are established by a set of the botnet. Using compromised PCs attacks the target machine while remaining anonymous.	DP flood, Slowloris, TCP flood, Zero-day DDoS, NTP amplification
2	Distributed reflective DoS (DRDoS)	Conventional systems can't control certain types of attacks that use legal hosts (reflectors) to assault the target machine with many response packets using forged IP addresses, Attack of the Smurfs, Attack of the Fraggles AF. The intruder transmits multiple requests to genuine nodes (reflectors) with a fake source IP address (the target address), which response with many voluminous messages to the fake IP (target server), overloading the target.	Smurf attack, Fraggle attack
3	Stealthy attack	Silently launched and stay unidentified by concealing the identity of the intruder's activities	Stealthy packet dropping
4	Physical attack	It is an attempt to cause damage to the physical elements (network).	Stoned Boot, Cold Boot attack, Evil Maid
5	Password attack	It tries to steal passwords and is notified by unsuccessful logins (brute force) over a shorter time.	Dictionary attack, phishing attack
6	Probe	It is completed before an attacker assaults a specific target (IPsweep, port sweep).	IPsweep, portsweep
7	User to Root	It is possible to get illegal access to local administrator rights by initiating as an ordinary underserved user. However, these attacks result in high consumption of time and money.	Loadmore, perl, Xterm
8	Remote to Local	Illegal access to a computer system via a micro-machine. By using a controlled neural network, a local assault can be detected.	FTP write, Warezmaster

3. Methodology

In this study, we perform a thorough analysis to enhance security and eliminate unnecessary features that negatively impact the network model. We develop a linear SVM combined with a fuzzy model, using efficient learning methods to identify features in an unsupervised way. The performance of this approach is compared with other traditional methods, and simulations are carried out in the MATLAB 2020 environment. The NSL-KDD dataset is used for training and testing to validate our model. Fig 1 shows the framework of the proposed model.

a. NSL-KDD

Tavallaee [22] recommended the NSL-KDD dataset in 2009 to address several issues found in the KDD-CUP'99 dataset. This dataset includes various attack categories, and it provides separate sets of records for training and testing. There are 127,973 records for training and 22,544 records for testing. The dataset contains 41 features: 35 continuous and 6 symbolic. These features are categorized into basic, content, and traffic features. The attacks are classified into four types: U2R (User to Root), R2L (Remote to Local), probing, and denial of service (DoS) attacks. The differences between the training and testing datasets create a realistic scenario for detecting intrusions.

Intrusion detection process

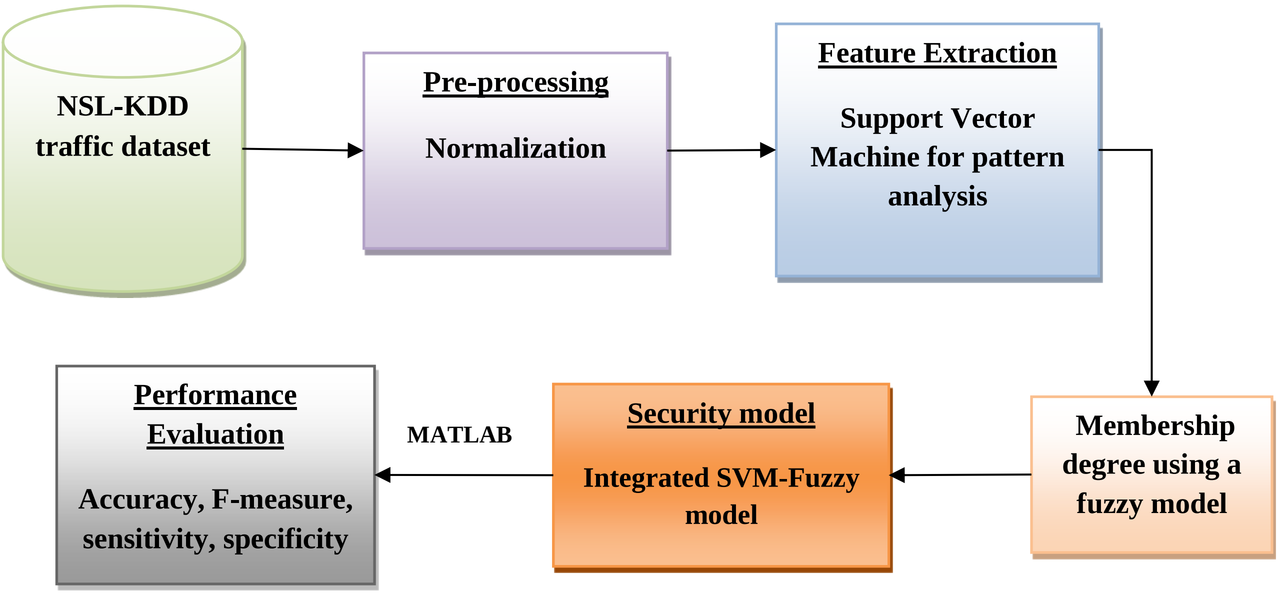
NSL-KDD traffic dataset
Pre-processing
Normalization
Feature Extraction
Support Vector Machine for pattern analysis
Membership degree using a fuzzy model
Security model
Integrated SVM-Fuzzy model
MATLAB
Performance Evaluation
Accuracy, F-measure, sensitivity, specificity

Figure 1. Block of the integrated fuzzy-SVM model

b. Pre-processing

In this process, the dataset normalization is applied to the non-numeric features such as flag, service, and protocol-type. These non-numeric features are transformed into numeric features using binary vectors. For example, the protocol-type feature might be converted to (0,0,1) for ICMP, (0,1,0) for UDP, and (1,0,0) for TCP. The service feature has 70 attributes, and the flag feature has 11 attributes. Normalization is done by computing the difference between the maximum and minimum values. This feature mapping is normalized using max-min normalization, which is mathematically expressed as shown in Equation (1):

$$x_i = \frac{x_i - min}{\max - min} \tag{1}$$

From Eq. (1), x_i represents a data point, max specifies the maximum values among all data points for each feature and min specifies the minimum values among all data points for each feature.

c. Fuzzy model

This section discusses the fuzzy model followed by SVM. Consider, $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_i, y_i)\}$ is a set of provided training samples where $x_i \in R^d$ and $y_i = \{-1, +1\}$ which specifies the i^{th} training samples and related target class. The samples are partitioned into two matrices, i.e. X_+^S and X_-^S where X_+^S and X_-^S are composed of samples with both negative and positive classes. For a non-empty set, the proposed fuzzy model is depicted below in Eq. (2):

$$A = \{(x, \mu_A(x))|_{x \epsilon X} \qquad (2)$$

Here, $\mu_A: X \rightarrow [0,1]$ and $\mu_A(x)$ specify the membership degree of $x \epsilon X$. The intuition fuzzy set is depicted as in Eq. (3):

$$\bar{A} = \{(x, \mu_{\bar{A}}(x), . v_{\bar{A}}(x))|_{x \epsilon X} \qquad (3)$$

Here, $\mu_{\bar{A}}(x)\ and\ v_{\bar{A}}(x)$ specifies the membership/non-membership degree of $x \epsilon X$ specifically; $\mu_{\bar{A}}: X \rightarrow [0,1], v_{\bar{A}}: X \rightarrow [0,1]$ and $0 \leq \mu_{\bar{A}}(x) + v_{\bar{A}}(x) \leq 1$, and $x \epsilon X$ is represented as in Eq. (4):

$$\pi_{\bar{A}}(x) = 1 - \mu_{\bar{A}}(x) - v_{\bar{A}}(x) \qquad (3)$$

The fuzzy model is depicted as $\alpha = (\mu_\alpha, v_\alpha)$ where $\mu_\alpha \epsilon [0,1], v_\alpha \in [0,1]$ and $0 \leq \mu_\alpha + v_\alpha \leq 1$. The largest fuzzy model is $\alpha^+ = (1,0)$ and the smaller fuzzy model is $\alpha^- = (1,0)$. The larger fuzzy model for the provided $\alpha = (\mu_{\alpha,} v_\alpha)$ is evaluated as in Eq. (5):

$$s(\alpha) = \mu_\alpha - v_\alpha \qquad (5)$$

Here, $s(\alpha)$ specifies the larger fuzzy model $\alpha = (\mu_\alpha, v_\alpha)$. Moreover, it is not probable to specify the larger fuzzy model score. To handle this issue, some functions need to be substituted and expressed as in Eq. (6):

$$h(\alpha) = \mu_\alpha + v_\alpha \qquad (6)$$

Based on Eq. (6) and Eq. (4) and it is expressed as in Eq. (7):

$$h(\alpha) + \pi(\alpha) = 1 \qquad (7)$$

If $s(\alpha_1) = s(\alpha_2)$ and $h(\alpha_1) < h(\alpha_2)$, then $\alpha_1 < \alpha_2$ and it is expressed as in Eq. (8):

$$H(\alpha) = \frac{1 - v(\alpha)}{2 - \mu(\alpha) - v(\alpha)} \qquad (8)$$

Eq. (8) shows the score function and the relationship among the membership and non-membership function and it is expressed as in Eq. (9):

$$s(\alpha_1) < s(\alpha_2) = H(\alpha_1) < H(\alpha_2) \qquad (9)$$

$$s(\alpha_1) = s(\alpha_2), h(\alpha_1) < h(\alpha_2) = H(\alpha_1) < H(\alpha_2) \qquad (10)$$

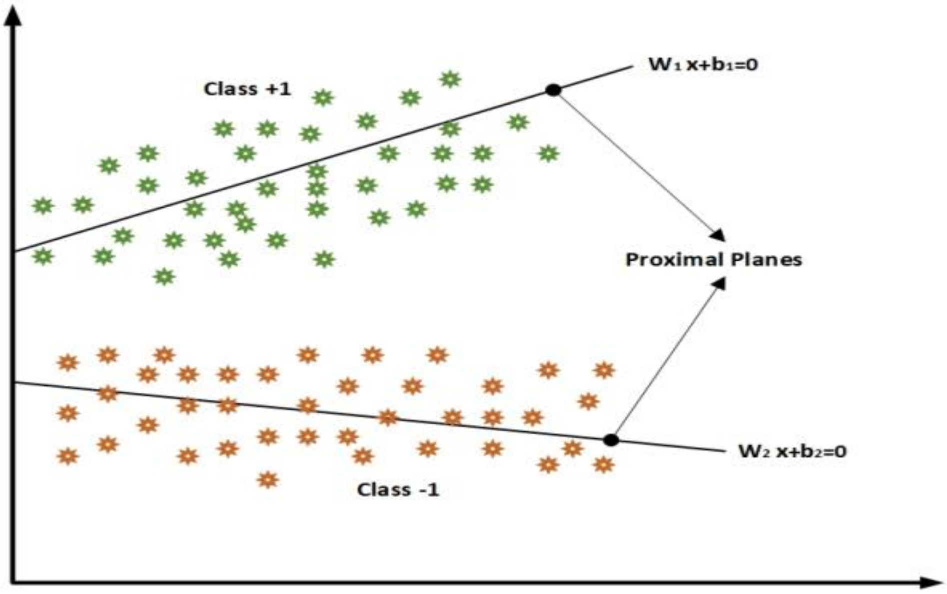

Figure 2. Generic SVM representation

d. Linear pattern analysis with SVM

Some conventional SVM is competent to resolve binary classification issues and intends to predict the optimal hyper-plane $w^T x + b = 0$ where $w\epsilon R^n$ specifies weight and $b\epsilon R$ specifies bias term (See Fig 2). The hyper-plane is utilized to determine the input sample label x_i and it is expressed as in Eq. (11):

$$\begin{cases} (w.x_i + b) \geq 0 & if\ y_i\ is + ve \\ (w.x_i + b) \leq 0 & if\ y_i\ is - ve \end{cases} \tag{11}$$

In linear SVM, an optimal hyper-plane is attained by resolving the quadratic programming issues in Eq. (12):

$$\begin{cases} \min \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i \\ y_i\ (w^T x_i + b) \geq 1 - \xi_i, \qquad \xi_i \geq 0, \quad i = 1,2, \dots, l \end{cases} \tag{12}$$

Where, ξ_i $(i = 1, 2, \dots, l)$, C, and l are slacking variables and penalty parameters with the number of training samples specifically.

e. Integrated SVM and fuzzy model for pattern analysis

Assume $\{(x_1, y_1, s_1), (x_2, y_2, s_2), \dots, (x_i, y_i, s_i)\}$ is considered as the set of training data composed of i samples with specific fuzzy memberships (s_i) where $\sigma \leq s_i \leq 1$ and $\sigma > 0$ is known as the smaller set of positive values. Let $z = \emptyset(x)$ specify the mapping from $\mathbb{R}^N$ represents feature space $\mathbb{Z}$. The optimal hyperplane is expressed as in Eq. (13):

$$\min \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i \tag{13}$$

$$y_i\ (w.z_i + b) \geq 1 - \xi_i; \qquad \xi_i \geq 0; \quad i = 1, \dots, l \tag{14}$$

Here, ξ_i specifies error identified in the linear SVM model and $s_i\xi_i$ specifies the error measured with various weighting and C specifies constant. The value of C specifies the reduced ξ_i efficiency and it is shown in Eq. (14). The lagrangian is evaluated to resolve the issues and it is shown in Eq. (15):

$$L(w,b,\xi,\alpha,\beta) = \frac{1}{2}w^T.w + C\sum_{i=1}^{l} s_i\xi_i - \sum_{i=1}^{l}\alpha_i(y_i(w.z_i + b) - 1 + \xi_i) - \sum_{i=1}^{l}\beta_i\xi_i \quad (15)$$

The following conditions need to be fulfilled to predict the saddle point $L(w,b,\xi,\alpha,b)$. It is expressed as in Eq. (16) to Eq. (18):

$$\frac{\partial L(w,b,\xi,\alpha,\beta)}{\partial w} = w - \sum_{i=1}^{l}\alpha_i y_i z_i = 0; \quad (16)$$

$$\frac{\partial L(w,b,\xi,\alpha,\beta)}{\partial b} = w - \sum_{i=1}^{l}\alpha_i y_i = 0; \quad (17)$$

$$\frac{\partial L(w,b,\xi,\alpha,\beta)}{\partial \xi_i} = s_iC - \alpha_i - \beta_i = 0; \quad (18)$$

Resolve Eq. (16) to Eq. (18) in Eq. (15) and it is re-written in Eq. (19) and Eq. (20):

$$maximize\ W(\alpha) = \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j K(x_i x_j) \quad (19)$$

$$\sum_{i=1}^{l} y_i\alpha_i = 0; \quad 0 \le \alpha_i \le s_iC; \quad i = 1, \dots, l \quad (20)$$

The above model needs to fulfill the vector condition and it is expressed as in Eq. (21) and Eq. (22):

$$\bar{\alpha_\iota}\left(y_i(\bar{w}.z_i + \bar{b}) - 1 + \bar{\xi_\iota}\right) = 0; \quad i = 1, \dots, l \quad (21)$$

$$(s_iC - \bar{\alpha}_i)\,\xi_i = 0; \quad i = 1, \dots ., l \quad (22)$$

Based on the above condition, the point x_i specifies the corresponding value $\bar{\alpha_\iota} > 0$ is termed as support vector. The integrated fuzzy and vector model possesses two kinds of support vectors. The first condition is $0 < \bar{\alpha}_i < s_iC$ which lies in the hyperplane margin and the successive one is $\bar{\alpha}_i = s_iC$ representing the misclassification process (Fig 3). On contrary, the integrated model predicts the points with $\bar{\alpha}_i$ into various kinds of support vectors based on s_i.

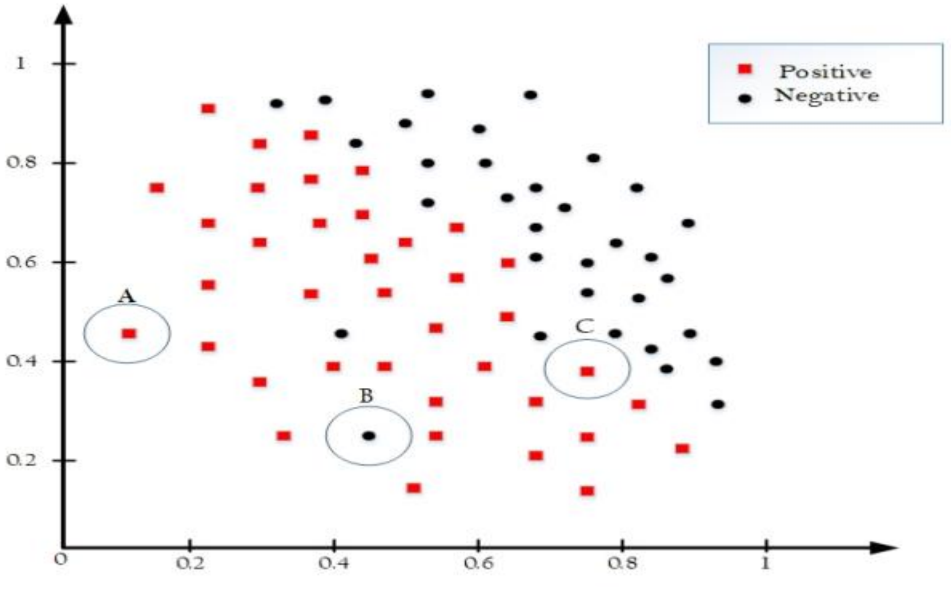

Figure 3. Predicted positive and negative samples

The integrated fuzzy and SVM model uses membership function and intends to diminish the outliers and noise which is appropriate chose the membership function. For instance, the training samples are provided in the boundary region with two different classes with the same membership degree. It may lead to wrong predictions. Here, the anticipated model adopts (μ, v) for handling this issue for training every sample where μ specifies the membership degree to one class and v specifies the non-membership degree function to other classes. However, the non-membership degree is connected with both positive and negative classes. The non-membership and membership degree for every training sample with high-dimensional feature space are given below:

1) The distance among the class center and training samples are utilized as membership function with higher dimensional feature space. For every training sample, the membership degree is expressed as in Eq. (23):

$$\mu\ (x_i) = \begin{cases} 1 - \dfrac{||\emptyset(x_i) - C^+||}{r^+ + \delta} & y_i = +1 \\ 1 - \dfrac{||\emptyset(x_i) - C^-||}{r^- + \delta} & y_i = -1 \end{cases} \tag{23}$$

Here, $\delta > 0$ is known as the adjustable parameter, r^+ (r^-) and $C^+\ and\ C^-$ specifies the radius and center of both positive and negative classes where $||.||$ specifies the distance among the input and corresponding class center. It is expressed as in Eq. (24):

$$D\ \left(\emptyset(x_i), \emptyset(x_j)\right) = \left\|\emptyset(x_i) - \emptyset(x_j)\right)\right\|| \tag{24}$$

Here, $\emptyset$ specifies the input samples with a high dimensional space vector. Similarly, the class center is provided as in Eq. (25):

$$C^{\pm} = \frac{1}{l_{\pm}} \sum_{y_i = \pm 1} \emptyset\ (x_i) \tag{24}$$

Here, l_+ and l_- specify the total amount of positive and negative samples. The functionality of the integrated SVM and fuzzy model is shown in Algorithm 1.

```
Algorithm 1:
Input: Training set $\{(x_1, y_1, s_1), (x_2, y_2, s_2), \dots, (x_i, y_i, s_i)\}$ where $C \in \{+, -\}$, larger and smaller membership function;
Output: SVM-based fuzzy membership function;
1. Partition the dataset into two sets and predict the samples as positive and negative samples;
2. Count the number of instances and record the samples as positive and negative;
3. Evaluate the imbalance class ratio;
4. Evaluate high-dimensional space vector based on positive and negative classes;
5. For every sample, evaluate the distance among the data patterns;
6. Evaluate the relative density among the samples and predict the outliers or noises among various classes;
7. For every sample, compute the membership function;
8. Train fuzzy-based SVM and initialize the parameters to evaluate the classification function;
9. End process;
```

4. Numerical results and discussion

This study uses the NSL-KDD dataset to test the effectiveness of the new model for improving network intrusion detection. The tests were run on an Intel Core i5 processor at 2.71 GHz with 8GB of RAM. The simulations were conducted in MATLAB 2020, with the kernel used as the classifier model. Different performance metrics were used to assess the SILF's performance, and the results from the training and testing phases were analyzed. The performance metrics are defined as follows: 1) True Positive (TP): Correctly identifying a threat as a threat; 2) True Negative (TN): Correctly identifying normal data as normal; 3) False Positive (FP): Incorrectly identifying normal data as a threat and 4) False Negative (FN): Incorrectly identifying a threat as normal. The metrics are then calculated based on these definitions.

1) Accuracy- specifies the appropriate proportion of total records in the NSL-KDD testing set can be specified mathematically expressed as in Eq. (16):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{16}$$

2) Precision- specifies the proportion of properly predicted intrusions to the total predicted intrusions during the testing process is known as the precision. It is mathematically expressed as in Eq. (17):

$$Precision = \frac{TP}{TP + FP} \tag{17}$$

3) Recall: specifies proportion of properly predicted intrusions to the total number of actual intrusion samples in the testing set is known as the recall or sensitivity. It is mathematically expressed as in Eq. (18):

$$Recall = \frac{TP}{TP + FN} \tag{18}$$

4) F-measure: It is the measure of both recall and precision as shown in Eq. (19):

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{19}$$

The experimentation is performed to examine the effectiveness and efficiency of the lower-level features provided to the $l - SVM$ classifier model with binary class $('0' - normal, '1' - \ anomaly)$ with the multi-classes $(probe, U2R, R2L, and\ normal)$ over the provided dataset. Moreover, training and testing are performed to compute the efficiency of the anticipated model.

Table 2: Prediction rate comparison with NSL-KDD dataset using various methods

Method	**DOS**	**R2L**	**Probe**	**U2R**
FC- ANN	86.05	83.18	48.12	83.33
TANN	89.94	80.53	84.89	60
SA-DT-SVMS	80	83.22	88.36	80
BPNN	80.35	89.12	89.12	25.58
GA-DBN	89.45	8.8	89.3	88.68
Proposed Method	**90.21**	**89.13**	**91.65**	**93.56**

Table 2 depicts the comparison of the prediction rate of the anticipated model with various approaches using the NSL-KDD dataset. The prediction of attacks like DoS, R2L, probe, and U2R using various existing approaches like FC-ANN, TANN, SA-DT-SVM, BPNN, GA-DBN, and integrated fuzzy and SVM is performed. The performance of the anticipated model over various existing approaches is substantially higher and establishes a better trade-off compared to others. While in the case of DoS prediction, the anticipated model gives 90.21% prediction accuracy which is 4.16%, 0.27%, 10.21%, 9.86%, and 0.76% higher than FC-ANN, TANN, SA-DT-SVM, BPNN, and GA-DBN. While predicting R2L, the anticipated model gives 89.13% prediction accuracy which is 5.94%, 8.59%, 5.9%, 0.01%, and 80.33% higher than FC-ANN, TANN, SA-DT-SVM, BPNN, and GA-DBN. In the case of probe attack, the anticipated model gives 91.65% prediction accuracy which is 43.53%, 6.76%, 3.29%, 2.53%, and 2.35% higher than FC-ANN, TANN, SA-DT-SVM, BPNN, and GA-DBN. In the case of U2R attack, the anticipated model gives 93.56% prediction accuracy which is 10.23%, 33.56%, 13.56%, 67.98%, and 4.88% higher than FC-ANN, TANN, SA-DT-SVM, BPNN, and GA-DBN (See Fig 4). The prediction of the attack over the network is achieved with optimal outcomes.

Table 3: Performance metrics evaluation

Method	**ACC**	**DR**	**Precision**	**Recall**	**FAR**
DOS	90.21	93.9	94.11	88.60	0.8
Probe	89.12	94.8	92.58	85.14	0.75
R2L	91.65	95.6	96.25	86.22	1.7
U2R	93.56	95.55	93.89	85.4	1.6

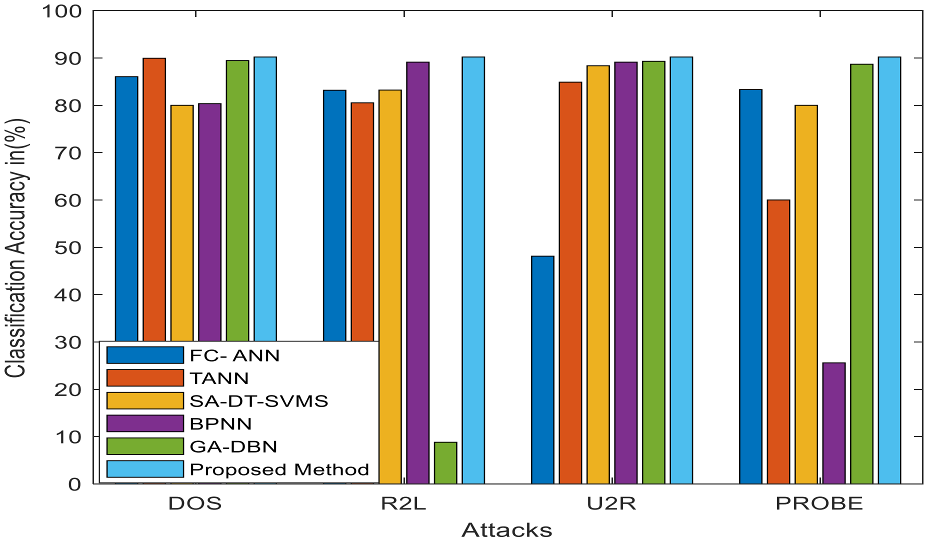

Figure 4. Evaluation of classification accuracy

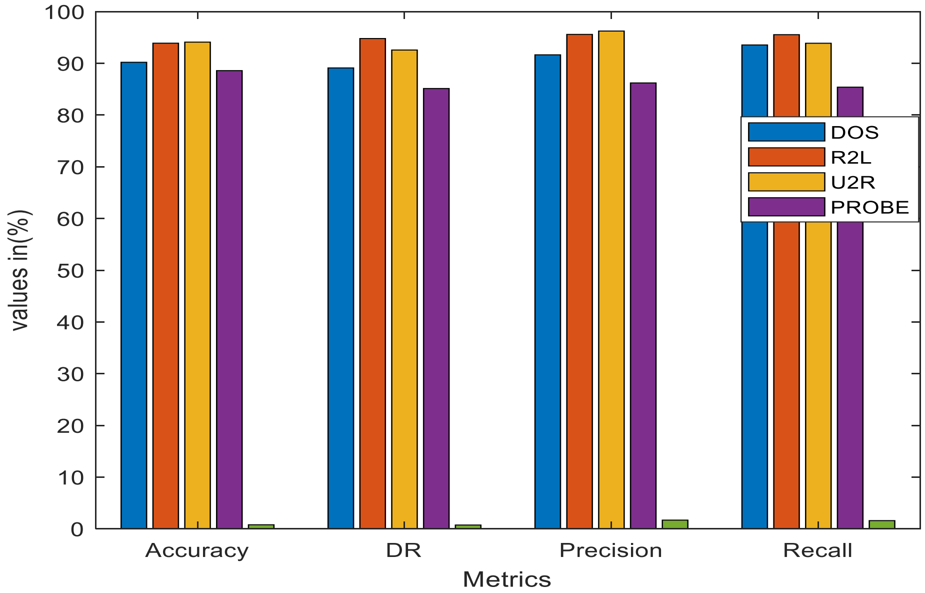

Figure 5. Performance metrics evaluation

Table 3 depicts the performance evaluation of the proposed model over the provided network traffic dataset, NSL-KDD (See Fig 5). The accuracy of the anticipated model for predicting DoS is 90.21%, the probe is 89.12%, R2L is 91.65% and U3R is 93.56%. The detection rate of DoS with the integrated fuzzy-SVM model is 93.9%, the probe is 94.8%, R2L is 95.6%, and U2R is 95.55%. The precision of the anticipated model for DoS prediction is 94.11%, the probe is 92.58%, R2L is 96.25% and U2R is 93.89%. The recall of the anticipated model for DoS prediction is 88.60%, the probe is 85.14%, R2L is 86.22%, and U2R is 85.4%. At last, the FAR of the anticipated model for DoS prediction is 0.8, the probe is 0.75, R2L is 1.7, and U2R is 1.6. It is proven that the model works effectually for predicting attacks over the network traffic.

5. Conclusion

With the adoption of SVM, the optimal particles are generation and the vector model effectively deals with the high-dimensional and complex data. The model intends to provide superior classification outcomes. Therefore, this work intends to provide the enhanced version with the integration of SVM with the fuzzy model and it is termed as a lightweight model. It is specifically for the IoT applications to overcome the complexity while detecting intrusions. The anticipated model considers an appropriate IDS for categorizing networks. Based on this, the problem of facing diverse attacks during the appropriate selection of IoT network models needs to be resolved. Thus, the model needs to enhance the classification accuracy with superior model generalization. The particle selection and membership integrated works to reduce the model complexity. However, some other advantages are also connected with this model. The provided network structure for certain attack types are superior to accuracy than other model and attains better results. The model constructed for certain attack types is higher in classification accuracy than other models. Even in the case of smaller training sets, the classification accuracy of the anticipated model is substantially superior to other approaches. Additionally, the complexity is diminished and training time is drastically diminished devoid of influencing the models' classification accuracy. Additionally, the model integrates fuzzy and SVM models which is not only considered for intrusion detection but also suited for other conditions like recognition and classification. For the various training set, the optimal network model has been generated adaptively during the classification process. The anticipated model shows a prediction rate of 90.21% for DoS attacks, 89.13% for R2L, 91.65% for probe, and 93.56% for U2R attacks. The major research constraint is the adoption of the existing network dataset for prediction purposes. However, with smaller training sets, higher classification accuracy is attained and reduces the lower frequency of attacks in the intrusion detection systems. In the future, the model considers the optimal amount of parameters over the deep network model and reduces the training time with improved prediction accuracy.

References

[1] Q. Jing, A. V. Vasilakos, J. Wan, J. Lu, and D. Qiu, "Security of the Internet of Things: perspectives and challenges," Wireless Networks, vol. 20, pp. 2481–2501, 2014

[2] S. Hemamalini ,V. D. Ambeth Kumar ,R. Venkatesan,S. Malathi. (2023). Relevance Mapping based CNN model with OSR-FCA Technique for Multi-label DR Classification. Journal of Fusion: Practice and Applications, 11 (2), 90-110.

[3] C. S. Manigandaa,V. D. Ambeth Kumar,G. Ragunath,R. Venkatesan,N. Senthil Kumar. (2023). De-Noising and Segmentation of Medical Images using Neutrophilic Sets. Journal of Fusion: Practice and Applications, 11 (2), 111-123.

[4] HaddadPajouh, A Dehghantanha, R Khayami, KK Choo, "A deep Recurrent Neural Network-based approach for Internet of Things malware threat hunting", Future Generation Computer Systems, vol. 85, pp. 88–96, 2018.

[5] Pajouh, R. Javidan, R. Khayami, D. Ali, and K. K. R. Choo, "A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IoT backbone networks," IEEE Transactions on Emerging Topics in Computing, vol. PP, no. 99, pp. 1–1, 2016.

[6] Kumar, V.D.A., Sharmila, S., Kumar, A. et al. (2023). A novel solution for finding postpartum haemorrhage using fuzzy neural techniques. Neural Comput & Applic. 35(33), 23683–23696

[7] Hodo, X. J. A. Bellekens, A. Hamilton, C. Tachtatzis, and R. C. Atkinson, "Shallow and deep networks intrusion detection system: A taxonomy and survey," CoRR, vol. abs/1701.02145, 2017.

[8] Sathya Preiya, V., and V. D. Ambeth Kumar. (2023). Deep Learning-Based Classification and Feature Extraction for Predicting Pathogenesis of Foot Ulcers in Patients with Diabetes. Diagnostics 13(12), 1983.

[9] Ma, F Wang, J Cheng, Y Yu, and X Chen, "A hybrid spectral clustering and deep neural network ensemble algorithm for intrusion detection in sensor networks," in Sensors vol. 16, no. 10, 2016.

[10] C Yin, Y Zhu, J Fei, and X He, "A deep learning approach for intrusion detection using recurrent neural networks," in IEEE Access vol. 5 pp. 21954-21961, 2017

[11] Hemamalini, Selvamani, and Visvam Devadoss Ambeth Kumar. (2022). Outlier Based Skimpy Regularization Fuzzy Clustering Algorithm for Diabetic Retinopathy Image Segmentation. Symmetry, 14(12), 2512.

[12] Abolhasanzadeh, "Nonlinear dimensionality reduction for intrusion detection using auto-encoder bottleneck features," in 2015 7th Conference on Information and Knowledge Technology (IKT), Urmia, Iran, pp. 1–5, 2015.

[13] Tan, W. Huang, and Q. Li, "An intrusion detection method based on DBN in ad hoc networks," in International Conference on Wireless Communication and Sensor Network, Wuhan, China, pp. 477–485, 2016

[14] Alghuried, "A Model for Anomalies Detection in Internet of Things (IoT) Using Inverse Weight Clustering and Decision Tree," Masters dissertation, Dublin Institute of Technology, 2017.

[15] S.M.H. Bamakan, H. Wang, Y. Tian, Y. Shi, An effective intrusion detection framework based on mclp/svm optimized by time-varying chaos particle swarm optimization, Neurocomputing 199 (2016) 90–102

[16] Ambeth Kumar, V.D. Vaishali,S. Shweta, B. (2015). Basic Study of the Human Foot. Biomedical and Pharmacology, 8(1), 435-444.

[17] Piyush K. Pareek, Pixel Level Image Fusion in Moving objection Detection and Tracking with Machine Learning “,Fusion: Practice and Applications, Volume 2 , Issue 1 , PP: 42-60, 2020

[18] Shivam Grover, Kshitij Sidana, Vanita Jain, “Egocentric Performance Capture: A Review”, Fusion: Practice and Applications, Volume 2, Issue 2 , PP: 64-73, 2020.

[19] Abdel Nasser H. Zaied, Mahmoud Ismail and Salwa El- Sayed, A Survey on Meta-heuristic Algorithms for Global Optimization Problems, Journal of Intelligent Systems and Internet of Things,Volume 1 , Issue 1 , PP: 48-60, 2020

[20] Kumar, I., Kumar, A., Kumar, V.D.A. et al. (2022) Dense Tissue Pattern Characterization Using Deep Neural Network. Cogn Comput 14, 1728–1751.

[21] Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Deep learning approach for network intrusion detection in software-defined networking," in Proc. Int. Conf. Wireless Netw. Mobile Commun. (WINCOM), Oct. 2016, pp. 258–263

[22] Tavallaee, E. Bagheri, W. Lu, and A. A. A. Ghorbani, ‘‘A detailed analysis of the KDD CUP 99 data set,’’ in Proc. IEEE Symp. Comput. Intell. Secure. Defense Appl., Jul. 2009, pp. 1–6.

[23] Bhattacharjee, A. K. M. Fujail, and S. A. Begum, ‘‘Intrusion detection system for NSL-KDD data set using vectorized fitness function in genetic algorithm,’’ Adv. Comput. Sci. Technol., vol. 10, no. 2, pp. 235–246, 2017.

[24] Martens and I. Sutskever, ‘‘Learning recurrent neural networks with hessian-free optimization,’’ presented at the 28th Int. Conf. Int. Conf. Mach. Learn., Bellevue, WA, USA, Jul. 2011, pp. 1033–1040.

[25] Kim, N. Shin, S. Y. Jo, and S. H. Kim, “Method of intrusion detection using deep neural network,” in Proc. IEEE Int. Conf. Big Data Smart Comput., Hong Kong, China, Feb. 2017, pp. 313–316.

[26] Paulauskas and J. Auskalnis, ‘‘Analysis of data pre-processing influence on intrusion detection using NSL-KDD dataset,’’ in Proc. Open Conf. Elect., Electron. Inf. Sci. (eStream), Apr. 2017, pp. 1–5.

[27] Garcia Cordero, S. Hauke, M. Muhlhauser, and M. Fischer, “Analyzing flow-based anomaly intrusion detection using replicator neural networks,” in Proc. 14th Annu. Conf. Privacy, Security. Trust, Auckland, New Zeland, Dec. 2016, pp. 317–324

[28] Kang and J.-W. Kang, “Intrusion detection system using deep neural network for in-vehicle network security,” PLoS One, vol. 11, no. 6, Jun. 2016, Art. no. e0155781.

[29] Wang, W.-D. Cai, and P.-C. Wei, “A deep learning approach for detecting malicious JavaScript code,” Security Commun. Netw., vol. 9, no. 11, pp. 1520–1534, Jul. 2016.

[30] Chang, W. Li, and Z. Yang, “Network intrusion detection based on random forest and support vector machine,” in Proc. IEEE Int. Conf. Comput. Sci. Eng./IEEE Int. Conf. Embedded Ubiquitous Comput., Jul. 2017, pp. 635–638.