Webology (ISSN: 1735-188X)
Volume 18, Number 6, 2021

Intrusion Detection On The Unsw-Nb15 Dataset Using Feature Selection And Machine Learning Techniques

J. Vimal Rosy[1*] and Dr. S. Britto Ramesh Kumar[2]

Research Scholar[1], Assistant Professor[2]

[1,2]St.Joseph's College (Autonomous), Affiliated to Bharathidasan University, Trichy, India.

Abstract

Backdoor, Exploits, Shellcode, analysis, fuzzers, generic, normal, reconnaissance, DoS, and Worms assaults are among the threats detected and classified by this research study using the UNSW-NB15 dataset. To improve accuracy, the feature selection process is carried out utilising the VFS (Validated Feature Selection) algorithm, which selects just the most important features. Finally, utilising the EC (Estimated Classifier) algorithm for classification, the incursion is classified. The sort of assault is revealed during the prediction phase. Finally, the new VFS-EC model was assessed using various performance measures and compared to other existing models to demonstrate its efficacy. The findings of the study revealed that this method is highly effective in identifying and classifying attacks with greater precision.

Keywords: -Intrusion detection, UNSW-NB15 dataset, Feature Selection, Estimated Classifier are some of the terms used in this paper.

I. Introduction

According to the Global Internet Statistics Report, the number of active users on the internet has increased to 4.66 billion in recent days, generating more than 2 quintillion bytes of data per day. It demonstrates that the velocity of data access from various sources has accelerated dramatically, as has the development of techniques and hacking tools. As a result, data security and privacy are required to protect data from various intrusions or hostile attacks. Because of the increased volume and speed of data, traditional intrusion detection systems were unable to detect assaults or intrusions in a timely and efficient manner. Certain computational procedures, on the other hand, are difficult by nature to handle such data, necessitating advanced intelligent approaches and strong technologies. Intrusion detection systems, or IDS, play a critical role in identifying attacks. The IDS system will monitor network traffic in order to detect threats, attacks, or suspicious activity. When this type of activity is detected, it may send an alert to the appropriate administrator. A variety of machine learning techniques can be used to efficiently deal with the infiltration. Different machine learning techniques can be used to efficiently manage and classify intrusions or attacks [1-3].

For more than two decades, IDS systems have been used to improve network and information system security, and it has been referred to as a significant tool [4]. In order to safeguard smart IoT devices, IDS is used, and it has handled numerous assaults while analysing and monitoring suspicious traffic in a networking environment [5]. Traditional IDS method execution on IoT is referred to as a trial due to the unique protocol stacks, standards, and architectural constraints. Because all attack types are ineffective at protecting, new solutions are required, such as physical hardware applications that use network probes to transmit encrypted data to a distant server and perform malicious detection. However, it necessitates substantial resources [6]. The IDS is critical in repelling hacker intrusion and producing effective IDS, which is referred to as a serious challenge. Machine learning algorithms can be used to detect suspicious attacks. Machine learning algorithms are developed and applied to undiscovered input during the detection process. In a network, several categorization algorithms, such as machine learning, are employed to detect assaults. Feature reduction strategies can be used to improve the detection time and the performance of classifiers [7].

The first intrusion detection system was created in 1980, and since then, various mature IDS products have emerged. However, several IDSs continue to have a high false alarm rate, generating several signals for non-threatening circumstances, which concerns security analysts because real malicious attacks can be overlooked in some cases. As a result, some academics are concentrating their efforts on building IDSs with lower false alarm rates and higher detection rates. Another difficulty with current IDSs is that they are incapable of identifying unknown attacks. Because network environments are always changing, new attacks and variants occur often. Enhanced IDS is used to detect unknown attacks, which is critical. The researchers consider machine learning methods when creating IDSs. Machine learning is a type of artificial intelligence that can extract useful information from large datasets. IDSs based on machine learning can achieve decent detection levels and consequently enough generalizability in detecting unknown attacks and variants once sufficient training data is acquired. Furthermore, IDS based on machine learning does not require much technical knowledge, making it simple to develop and design. [8]

Certain comparative studies have been carried out, but no comprehensive research has yet been carried out. As a result, the focus of this research is on developing an IDS for networks with improved feature selection and classification methodologies, as well as investigating various and effective feature selection approaches. The current research focuses on combining feature selection with classification algorithms to improve IDS accuracy. Establishing the makeshift IDS presents some difficulties. Traditional IDS system execution is considered a trial due to architectural constraints, protocol stacks, and standards. Another aspect to consider while detecting intrusion is resource and computational limits. All sorts of attacks must be detected, and this will raise the rate of intrusion detection accuracy. A better feature dataset must be used to investigate all types of performance measures.

A unique VFS-EC algorithm is proposed to handle the above complications, such as focusing on all sorts of attacks and improving the accuracy of identifying incursion rates.

As a result, the study's main contribution is:

• The created IDS system should be effective in detecting all types of attacks quickly and accurately. To achieve this, cross verified Artificial Neural Networks with Random Forest Classifiers, namely EC, were utilised for classification, and an efficient feature selection approach, namely the VFS algorithm, was applied to boost classification accuracy. It also returns the best solution from the global optimum.

• The UNSW-BoT Dataset is used to analyse categorization performance. To assess the proposed IDS system's effectiveness in detecting malicious attacks, the performance was compared to that of other existing models.

1.1 Paper Organisation:

The rest of the paper is arranged as follows: part II discusses related IDS works as well as various machine learning methods for handling various assaults. In section III, the proposed VFS-EC model is briefly discussed. In addition, the results and debate are displayed and illustrated in section IV. Finally, future work is discussed in section V of the study.

II. Related Works

Two machine learning algorithms, feed forward neural network and boosted decision tree, have been established for detecting malicious behaviour in data transmission networks. Performance and sensitivity values are satisfactory. The results are analysed and compared to previous investigations, demonstrating their efficiently [9]. A deep learning-based approach was built in [10] to ensure standard security issues like as authenticity, trust, and privacy. In addition, this [11] model included a hybrid data optimization technique that included feature selection and data sampling. Isolation forest-iForest has been used to eliminate outliers, and a genetic algorithm-GA has been used to optimise the sampling ratio. The ideal training dataset was achieved using a random forest classifier on the UNSW-NB15 dataset. This model has been shown to have a few unusual behaviours. More time is necessary for data optimization, and online procedure assistance is considered a limitation. This methodology can also be used in other areas of anomaly detection, such as fraud detection. Because classifier training requires more time, searching methods are optimised.

Only one strategy is utilised for pre-processing data in all machine learning processes. Even while the random forest classifier takes longer, it performs better [12]. On the UNSW-NB15 and NSL-KDD datasets, as well as a network log in campus with 300 million daily records, another ensemble learning-based support vector machine, auto-encoder, and random forest were employed. The results of this model, when compared to the findings of other studies, reveal that the ensemble learning model reduces the number of incorrect negative and positive predictions [13]. IDS-CNN, a Convolutional Neural Network-based system, displayed tensor flow, traffic analysis, and a packet capture interface, among other open-source tools. The machine learning interface tensor flow focuses on neural network training and testing, intrusion response, and pre-processing. When compared to previous techniques, precision results reveal higher values [14].

On the KDD cup dataset, IDS based CNN had a higher detection rate when compared to other IDS classifier systems. The rate of false alarms should be kept to a minimum [15]. Alarm filtering has been focused on improving detection rate accuracy. In comparison to unsupervised clustering, a higher intrusion detection rate was achieved utilising hybrid ant colony optimization, which resulted in a lower false alarm rate. The ant colony optimization algorithm [16] was converged using K-means clustering. RNN-IDS stands for Recurrent Neural Networks with other machine learning models that were studied and their performance evaluated on the NSL-KDD dataset. Accuracy can be improvised, but the time spent on training is more important to consider [17]. The detection rates of unknown assaults have been improved utilising the support vector machine and extreme learning in multilayer-based hybrid IDS. The IDS performance and total training time were improved using the modified K-means algorithm [18].

Several optimization approaches were applied to the NSL-KDD dataset in order to maximise the rate of correctness. The invaders network protocol has allowed anomalies in network traffic to be spotted and evaluated using the NSL-KDD dataset [19]. The dimensionality reduction plays a significant role in higher dimensional network traffic since anomaly detection resulted in time consumption in higher dimensional network traffic. On the KDD CUP 99 dataset, a Bayesian network was used for classification, and relevant features were chosen using the firefly technique. Accuracy improved as well [20]. Furthermore, the accuracy of IDS, which is based on CNN and a random forest classifier that extracts features from raw network packets, has been improved. On organised data, random forest performed better, whereas unstructured data was handled by CNN. Certain attacks have been identified, but they have resulted in increasing computing complexity and length [21].

Due to resource limits and computational complexity, intrusion detection can be difficult in some instances. K-NN and SVM developed light-weight access control techniques to preserve computing resources and system lifetime in IoT devices [22]. In some circumstances, utilising data mining methodologies, the accuracy rate is not adequate, so a new IDS-based linear correlation co-efficient for feature selection was used, along with a conditional random field and CNN for classification. Greater precision has been achieved. On the basis of optimised conventional approaches, more efficiency has been achieved [23]. The efficiency of using traditional IDS-based advanced attack detection is lower. Deep learning models can also be used to handle this problem however they are mostly focused on Denial of Service (DoS) attacks. Probing attacks, DoS attacks, user to root U2R attacks, and remote to local attacks were all included in the KDD cup-99 dataset for testing. In comparison to RNN, CNN is more effective in detecting intrusions. There hasn't been much work on multi-class categorization yet [24]. For DoS attack detection, many machine learning and deep learning algorithms have been examined, with greater accuracy [25].

Another problem is to find different IoT malware detection methods that can work quickly and accurately in the IoT industry. The malware detection dynamic evaluation based neural network was used to extract different behaviours in relation to system calls, memory, and virtual process systems. It can also be converted into virus images, and the losses can be reduced by analysing these behaviour photos [26]. In automated IDS systems, U2R, probing attacks, and R2L assaults have been prioritised, and computational overhead has resulted in the detection of other attacks from NSL-KDD datasets. Stability and potentiality, on the other hand, were also noted [27]. For dimensionality reduction, an ELM-based neural network with extreme learning has been used. On the NSL-KDD dataset, both IDS execution and detection efficiency improved. There has been evidence of improved performance [28]. The paper uses the ML algorithm Canny method and the Smallest Univalue Segment Assimilating filter to see how effective Deep Learning performs. Additionally, using these SUSAN filters helps reduce image noise and helps detect the corner of images using the enhanced canny algorithm. SUSAN beats traditional edge detectors and gives more accurate results [29]. The ML algorithm using this proposed methods use the pre-trained neural network to recognise additional edge pixel values. A CNN is used to calculate the edge of an image patch. [30].

II. Proposed Methodology

The novel proposed IDS model, VFS-EC , has been elaborated in this part, and the entire flow is depicted in fig.1. The UNSW-NB15 dataset is first loaded, and then pre-processing is carried out. To improve accuracy, the feature selection process is carried out utilising the VFS algorithm, which selects just the most important features. Finally, utilising the Novel EC algorithm, the anomalies are classified. The sort of assault is revealed during the prediction phase. Finally, the new VFS-EC model was assessed using various performance measures and compared to other existing models to demonstrate its efficacy.

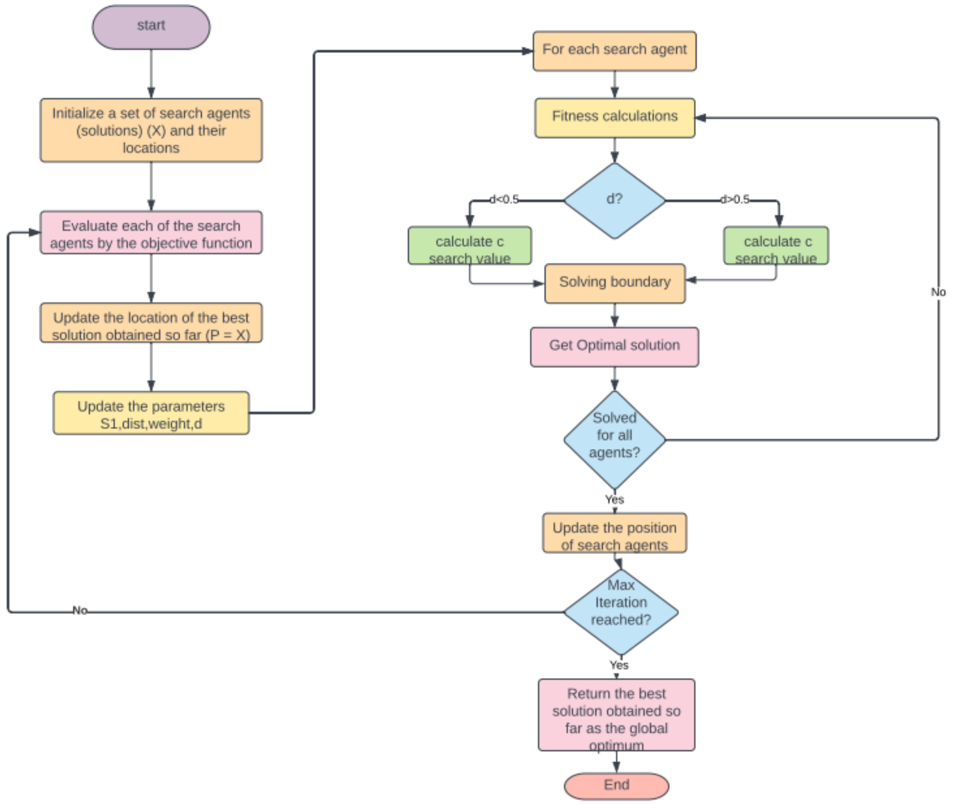
start
Initialize a set of search agents (solutions) (X) and their locations
Evaluate each of the search agents by the objective function
Update the location of the best solution obtained so far (P = X)
Update the parameters S1,dist,weight,d
For each search agent
Fitness calculations
d?
d<0.5
d>0.5
calculate c search value
calculate c search value
Solving boundary
Get Optimal solution
Solved for all agents?
No
Yes
Update the position of search agents
Max Iteration reached?
No
Yes
Return the best solution obtained so far as the global optimum
End

Fig. 1. Flow of the proposed VFS-EC Model

Here, “1” is represent feature is selected and second “0” is represents feature is discarded.

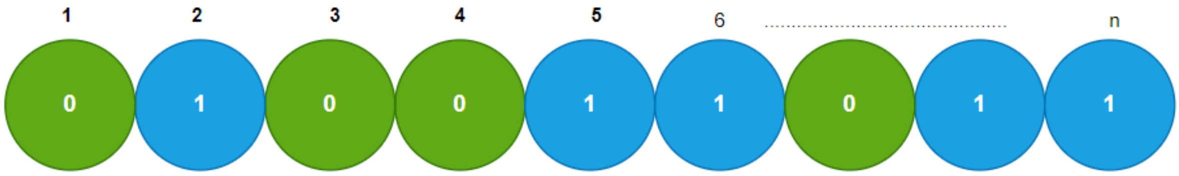

fig 2. feature selection

3.1 VFS Algorithm for Feature Selection

In this section, we present a detailed description of the proposed method. VFS is a process of selecting relevant features of a dataset in order to improve the learning performance, decreasing the computational complexity, and building a better classification model. Based on the nature of the feature selection problem, a VFS algorithm is usually applied to find an optimum feature subset. Every algorithm is represented as a vector with N entries, where N is the total number of features in a dataset. Each vector has the value 0 or 1, where zero indicates that the feature is not selected whereas one represents that the feature is selected.

we proposed to reduce the two formulas of sine and cosine into only one formula: Instead of switching between two formulas, we suggest to limit the updating position to only one of the previous formulas. Indeed, we believe that the choice between the two formulas is arbitrary because both sine and cosine functions return values between 1 and +1 for a real input in the interval [0, 2π]. Therefore, we think that Exemption from alternation between these two functions will not inflict the performance of the algorithm. On the other hand, to reach a good tradeoff between intensification and diversification, we introduce a new equation that depends on the random variables C and r1. This ensures an efficient transition between exploration and exploitation. Eq. (1) below shows the new update positions:

$$X(i,j)_{t+1} = \quad 1- [s-t^* (\frac{s}{\max\ iter})] * X(i,j)_t + [s-t^* \frac{s}{\max\ iter}] * \gamma \qquad \text{if } C < s1$$

$$\text{where } s1=s=t^*s/\max\text{ iter}$$

$$P_1(j)_t + [s-t^* \frac{s}{\max\ iter}] * \sin(dist) * |weight * P_1(j)_t - X(i,j)_t| \quad \text{if } C>= s1 \qquad \textbf{Eq .1}$$

where C = b*d, and b is a constant integer, generally in the interval [1, 5].

Where..

- S1=next positions region (movement direction)
- dist ∈ [0, 2π], is a random variable, difference between the current position and its previous position in the state space.

- weight is a random variable. d is used to choose different search paths, sine or not, according to different random values. gives weight to make d>1 or d <1 to reach its destination.
- Pi is the objective solution

```
Algorithm 1: VFS (agent no, dataset, dimension, max_iter,upper bound, lower bound)
INPUT:

Search agent number, dataset, dimension, max _iter, upper bound, lower bound

OUTPUT: Vector with 20 best solutions

1.Initilaize positions of all agents(X)
2.Calculate the cost function of each agent(fit), and select the best position's agent (Best_pos)
3. T-2; WHILE (t <= max_iter) DO s – t*(s/max_itera); % decreasing iteratively
FOR each agent (X), from the dimension (j) DO dist– (2*pi)*rand (); weight-rand (); C– b*rand (); % updating of the position of the solution
IF (C< s – t*(s/max_iteration) m=Get_rand_position(); %get a random position from the search space [lb,ub]  X(j)=(1-$r_1$)*  X(j)+($r_1$)*m;
ELSE  X(j)=Best_pos (j) + s – t*(s/max_iteration *(sin(dist))*abs(weight* Best_-pos(j)-X(j)); ENDIF % end of the updating solution's position
 END FOR
1. Calculate the new cost function (Fit) of each agent (X), and get the best position's agent (Best_pos).
2. Increment (t);

END WHILE END ALGORITHM
```

Algorithm 1 shows the pseudo-code of the VFS algorithm. In the first step, positions of solutions are randomly initialized in the search space. After that, a loop containing max iteration steps is executed to find an optimal feature subset. In each iteration of the loop, the score (Fit) of each solution is calculated using the cost function defined in Eq (3). After this evaluation phase, the best solution (Best pos) from the population is determined according to the best score. The updating of solutions is achieved using Eq (1). Particularly, for updating its position, each agent switches between two formulas that serve to the diversification and the intensification strategies respectively. This switching is controlled by two parameters s1 and d, where s1 decreases at each iteration and d is a random number between 0 and 1: The diversification is performed if the inequality (d < s1) holds, otherwise the intensification is performed. Moreover, the updating parameters (s1,d) are adjusted in order to obtain a suitable transaction from the diversification strategy to the intensification strategy, which ensures a good balance between them.

The variables used are the same features number in the given dataset. The variables are limited in the range [0, 1], where the variable value approaching 1 means that its corresponding features are candidate to be selected in the classification. In individual fitness calculations, the variable is the threshold to decide the exact features to be assessed as in Eq. (4):

fij = 1, if Xij > 0:5; otherwise, 0 Eq. (4)

where Xij is the value of dimension for search agent i at dimension j. While updating each position of search agent at some dimensions, the updated value can violate the limiting constraints: [0, 1]; hence, we use simple truncation rule to ensure variable limit. Each candidate feature is represented as a binary vector with one dimension. The vector is used for features mapped to be in [0, 1] interval based on threshold value that is set to 0.5, indicating to the upper bound (ub = 1) and the lower bound (lb = 0).

Initial population

The VFS Algorithm begins first randomly with positions to converge the global optima. It then calculates the value of fitness for every individual. It allocates the utmost remarkable location to FS as candidate features. Every solution is presented as a binary vector in one dimension. The number of the vectors is equal to the number of features in the original dataset. All cells within the vectors are labelled with 0 or 1. One value is indicating that the feature is selected, otherwise indicating that the feature is ignored.

Feature subset representation

A potential solution (a features subset) is represented by a features vector where each feature corresponds to a dimension and every variable is fixed to a range within [0, 1]. We assume that m-dimensional data set is input data to Feature selection technique. The data values are represented by matrix (Data n*m) where n represents total number of data samples and m represents total number of features in data set. The aim of Feature selection technique is to select optimal subset of features from all available features. Suppose X={X(i)| i=1,2, 3,..,m} is an original feature set with m dimension. The decision of selecting or rejecting a feature is taken as follows: if the position value is greater than or equal to 0.5, then the corresponding feature is selected. otherwise, this feature is rejected. In individual fitness calculation, a threshold is used to decide the exact features to be evaluated as in the following Eq. (2)

$$f_{i,j} = \begin{cases} 1 & \textit{if } Xi,j \geq 0.5 \\ 0 & \textit{otherwise} \end{cases}$$

here Xi,j is the dimension value for search agent i at dimension j. Notice that while updating each search agent’s position at some dimensions, the updated value can violate the limiting constrains [0, 1]. To overcome this problem, we have used a simple truncation rule to ensure variable limits.

Classifier

K-Nearest Neighbor (KNN) classifier is a predictor of variables weight at a distance based on trial-and-error process. K-Nearest Neighbor is utilized as a part of the fitness function in all the experiments due to its excellent performance in classifying.

Fitness Function

In this study, the fitness function is applied to assess each feature subset in the search space of Sine Cosine Algorithm based on K-Nearest Neighbor (KNN) as a classifier, where K = 5. The proposed fitness function is calculated by using Eq. (3).

$$\text{Fitness} = S * 0.01 * E_R(D) + C * \frac{|M|}{|N|} \qquad \text{Eq .3}$$

Feature selection can be considered as a multi-objective problem in which two contrary objectives must be satisfied. These two objectives are the maximum accuracy, and the other is the minimum number of selected features. The fitness function that is used to evaluate each individual is shown in Eq.3. where $E_R(D)$ is the classification error, D is the number of selected features constant value (fixed 0.05) to control the classification accuracy performance to the features number that is selected, N is the total number of features in the dataset, α and β are two parameters related to the importance of accuracy and number of selected features, $\alpha \in [0, 1]$ and β=1-α.

Mutation operator

The mutation operator can be employed with distinct traits that are not present in the ancestor to generate a new set of people. It can also be used for binary, real, or integer representations, and it comes in a variety of kinds. Mutations were created by randomly picking one or more bits and flipping the value depending on a set of probabilities. After using the crossover operator, the SC algorithm performs mutation as an internal function, which further builds new solutions and improves the exploration ability. The mutation operator is written as follows:

$$P_c^{v+1} = \text{Mutation}(P_c^v) \qquad \text{Eq. 4}$$

The suggested study used the VFS algorithm for feature selection, as illustrated in fig.2. The location is initialised, the objective function is used to evaluate the search agents, the best solution is changed, and the location is updated in this operation. The position of the search agent is also updated. Returning to the global optimum may be the best solution. In this approach, the important trait has been chosen.

3.2The EC method was used to classify the identified intrusion.

This ANN model is paired with a random forest classifier and k fold cross validation. This suggested model eliminates hyper parameter adjustment, making it suitable for large data volumes and low memory needs.

The random forest classifier's skill was assessed using the cross-validation approach. When evaluating the model's skill, the resampling procedure is taken into account. The K parameter indicates the number of groups created by splitting the dataset. Furthermore, the EC technique is used for accurate intrusion categorization, and this algorithm is more robust. The EC pseudo code is the algorithm that follows. The random forest classifier is made up of numerous decision trees that form a decision tree based on a data sample that predicts everyone and then votes on the best option. It done much better than the decision tree.

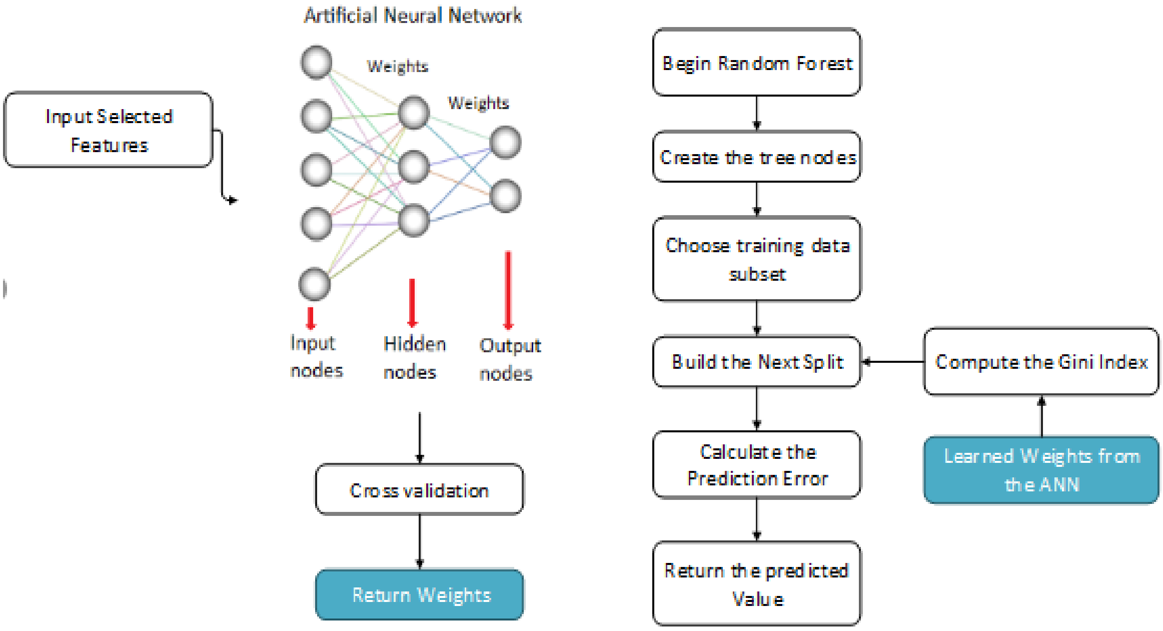

Fig 4 EC algorithm

Stopping Criteria

Without particular stopping criteria the feature selection process continuous its execution through the search space. Generation procedures and evaluation methods have different impact on the choice for a stopping criterion. Stopping criteria based on a subset generation includes: (i) maximum number of features, and (ii) maximum number of generations reached. Stopping criteria based on an evaluation method includes: (i) further process not produce a better subset and (ii) optimal subset according to evaluation methods is obtained. A process of feature selection halts by returning best selected subset of features to a machine learning algorithm. Generalized algorithm of Feature selection is shown below: Symbols: D= data set S= search technique M= Evaluation Measure Sc= stopping criteria F optimal= optimal subset of features.

Procedure Feature Selection (D,S, M,Sc)

REPEAT

(i) Generate subset of features (Ftemp) with subset generation procedure by using search procedure (S).

(ii) (ii) Validate subset (F temp) using evaluation technique M. If (Evaluation criterions are satisfied) Update F optimal with F temp. Until stopping criteria Sc is matched Return F optimal (optimal subset of features.)

IV. Results and Discussion

On the UNSW-NB15 dataset, the suggested VFS-EC intrusion detection algorithm performs well. The data was gathered from https://research.unsw.edu.au/projects/unsw-nb15-dataset. UNSW Canberra cyber range lab created the UNSW-NB15 dataset raw network packets for synthetic current behaviour attack and generating activities of real modern normal. UNSW Canberra cyber range lab created the UNSW-NB15 dataset raw network packets for synthetic current behaviour attack and generating activities of real modern normal.

4.1 Analyzed performance

The results of the proposed system's K-Fold cross validation are provided in table-1. Various types of cross validation have yielded different accuracy rates. 1 fold cross validation revealed 68.11 percent, 2 fold cross validation revealed 99.84 percent, 3 fold cross validation revealed 99.85 percent, and so on. The 5 fold cross validation, on the other hand, revealed a maximum accuracy rating of 99.87 percent.

Table-1. Cross-validation using the K-fold method

Method	Accuracy
1 fold cross validation	68.11
2 fold cross validation	99.84
3 fold cross validation	99.85
4 fold cross validation	99.86
5 fold cross validation	99.87

Table-2 also shows the total number of columns and the selected columns of the introduced system. From the results, it appears that almost 18 columns have been chosen out of a total of 40. The 18 columns chosen were determined to be useful.

Table-2. Columns from the dataset that were chosen

Total Number of Columns	40
Number of Selected Columns	2,5,6,8,9,10,12,16,18,20, 24,29,31,33,34,36,40

Performance of the Enhanced Classification Algorithm

The confusion matrix is generally utilised to assess the classifier model's performance. Obtained results are shown in fig.3. From the results, it is found that, **51821** have been correctly identified as attacks (True Positives), while, **117156** have been correctly detected to be normal (True Negatives). On contrary, **3613** attacks have been misinterpreted as normal (False Positives), whereas, **998** normal have been misclassified as attacks (False Negatives). Though the correct classification rate is maximum in comparison to the misclassified rate, the proposed classifier is found to be effective.

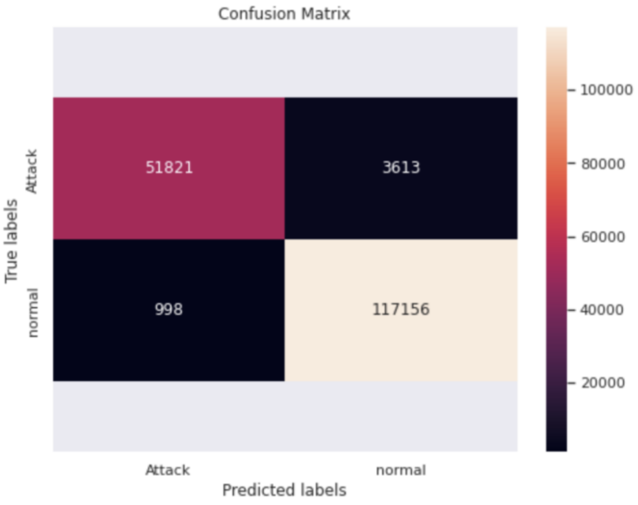

Fig.3. Matrix of ambiguity

4.2 Analyses Comparative

The suggested system's performance is compared in terms of precision, recall, F-measure, FPR (False Positive Rate), and accuracy. Table-3 displays the obtained results. The present methods for analysis are abnormal, normal, and weighted average. Figures 2 and 3 show the graphical outcomes of the study.

Table-3. In terms of performance measures, comparative analysis [29]

Method	Precision	Recall	F-measure	FPR	Accuracy
Abnormal	0.9471	0.9649	0.9561	0.0644	0.9515
Normal	0.9556	0.9353	0.9454	0.0353	0.9515
Weighted average	0.9516	0.9515	0.9515	0.054	0.9515
VFS-EC	0.9915	0.9915	0.9915	0.0256	0.9987

According to the findings, the normal technique had the highest precision rate of 0.9556 percent, but the proposed VFS-EC (k-fold Cross Validated Artificial Neural Network Weighted Random Forest Classification) method had the highest precision rate of 0.9915 percent. In the same way, abnormal method had the highest recall rate of 0.9471 percent. However, with 0.915

percent, the introduced technology outperforms previous methods. Similarly, the F—measure and accuracy rate of the procedures under consideration differed in each situation. The proposed work, on the other hand, yielded better results than traditional methods. On the contrary, a system's effectiveness is confirmed by its minimum FPR rate. As a result, as demonstrated in fig. 4, the suggested work had a lower FPR than existing approaches.

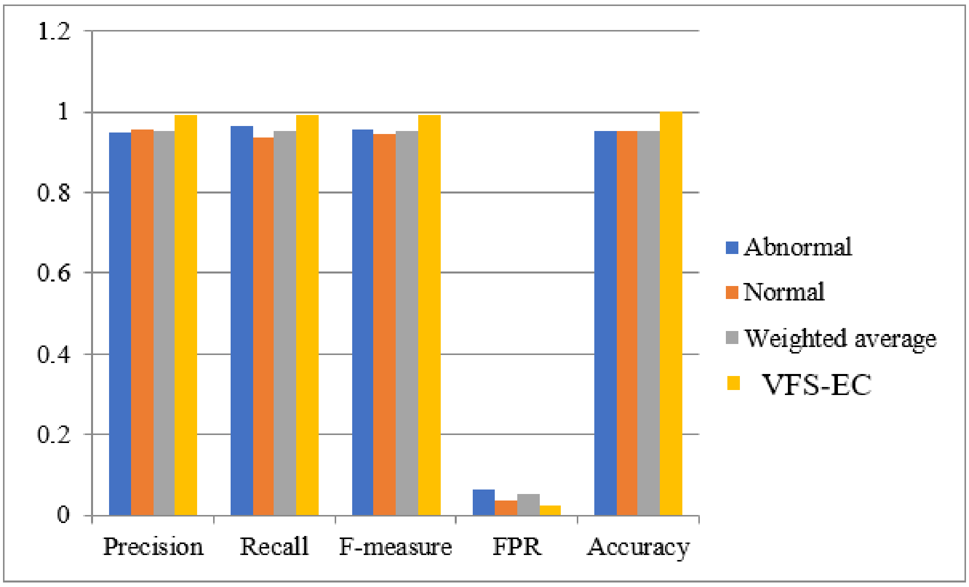

Fig.4. In terms of performance measures, comparative analysis [29]

Table 4 shows the comparison between the best transfer models with ML algorithms [32] LOGNN (Logarithmic Neural Network), RF (Random Forest), KNN (K Nearest Neighbour), CNN (Convolutional Neural Network), RNN (Recurrent Neural Network), DNN (Deep Neural Network), LSTM (Long Short-Term Memory), SVM rbf, Conv LSTM SAE NN, SDAE ELM1, SDAE ELM2, SDAE ELM3, stacked CNN LSTM.[34] Table-4 displays the obtained results. Traditional stacked CNN LSTM SAE NN and Conv LSTM SAE NN accuracy rates were 0.954 percent, which was higher than other current works. However, the proposed work was more accurate than the approaches explored, demonstrating its efficacy. Similarly, when compared to traditional studies, the precision, recall, and F1-score of the proposed system revealed improved outcomes.

Table-4. Analyzed in terms of performance metrics [30],[35]

Algorithm	Accuracy	Precision	Recall	F1-Score
LOGNN	0.941	0.95	0.946	0.948
CNN	0.589	0.578	0.935	0.717
DNN	0.759	0.819	0.724	0.766
LSTM	0.805	0.886	0.745	0.811
RNN	0.881	0.87	0.927	0.897

RF	0.905	0.986	0.866	0.923
LR	0.745	0.951	0.652	0.774
KNN	0.808	0.931	0.777	0.847
DT	0.896	0.98	0.863	0.918
SVM_rbf	0.655	0.997	0.491	0.658
Stacked_CNN_LSTM_SAE_NN	0.95	0.965	0.951	0.957
ConvLSTM_SAE_NN	0.951	0.962	0.951	0.957
SDAE_ELM1	0.739	0.727	0.844	0.781
SDAE_ELM2	0.771	0.773	0.82	0.796
SDAE_ELM3	0.79	0.775	0.88	0.822
DBN_EGWO_KELM	0.935	0.946	0.941	0.941
Proposed VFS-EC	0.9987	0.9915	0.9915	0.9915

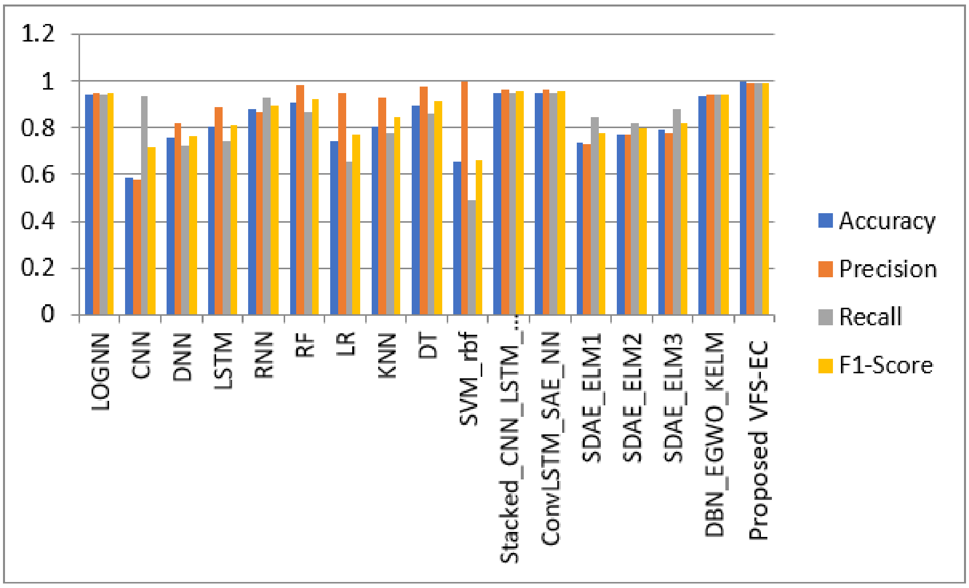

Fig.5. In terms of performance measures, a comparative analysis [30],[34] was conducted.

In addition, the proposed methodology's performance has been compared to that of traditional research [31]. For this analysis, accuracy, F1-score, recall, time, and precision were taken into account. Despite the fact that existing strategies produced greater results, the suggested methodology demonstrated effective performance across all measures. Furthermore, the new work's execution time was determined to be low, demonstrating its usefulness over traditional methods[36]. Figures 6 and 7 demonstrate the graphical depiction.

Table-5. Analysis with respect to various performance metrics [31]

Model	Metrics				
	Accuracy	Precision	Recall	F1 Score	Time(s)
MLP-1	98.97	93.364	98.45	95.84	430.94

MLP-2	98.98	93.6	98.37	95.93	595.01
MLP-3	98.97	93.21	98.54	95.81	606.22
MLP-4	98.9	92.65	98.66	95.56	617.16
CNN-1	99.11	95.51	97.73	96.37	423.57
CNN-2	99.11	94.7	98.07	96.37	892.18
RNN-1	98.9	93.61	97.65	95.62	128.93
RNN-2	98.81	92.21	98.28	95.14	273.46
RNN-LSTM-1	98.93	93.94	97.56	95.74	301.55
RNN-LSTM-2	98.87	92.62	98.12	95.28	388.28
RNN-GRU-1	98.87	93.2	97.86	95.47	317.03
RNN-GRU-2	98.8	92.11	98.37	95.15	412.74
CNN+LSTM-1	99.12	95.92	97.126	96.55	1355.7
CNN+LSTM-2	99.11	95.96	97.07	96.51	1334.2
Proposed VFS-EC	**99.87**	**99.15**	**99.15**	**99.15**	**126.3**

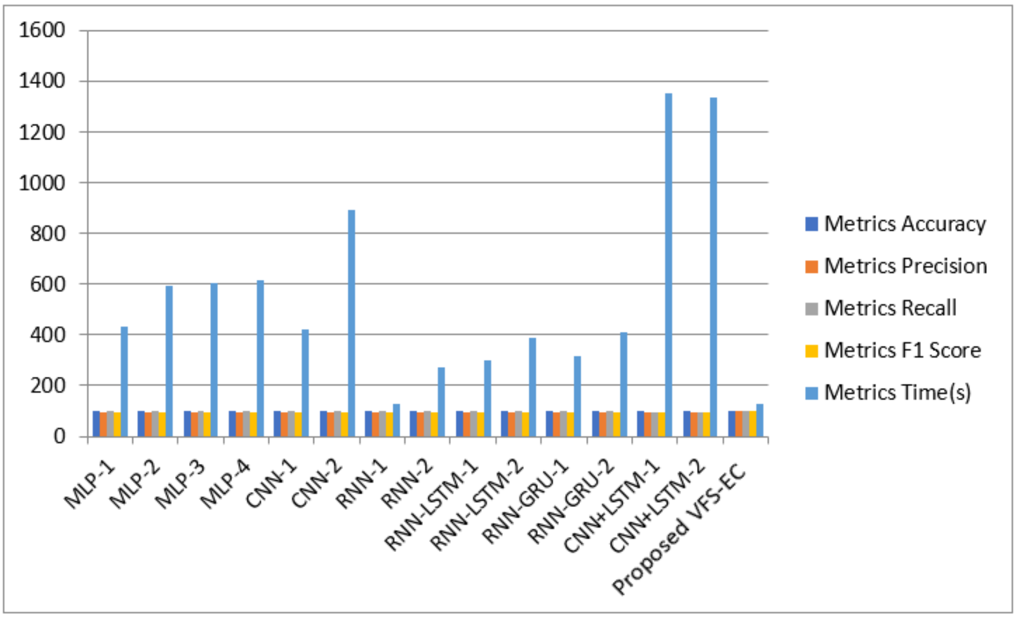

Fig.6. Comparative analysis with respect to performance metrics [31]

In terms of the metrics evaluated, the analytical results demonstrated that the proposed approach is more effective and efficient than traditional methods. The introduced work has a low rate of misinterpretation and a short execution duration, demonstrating its efficacy. As a result of these findings, it is suited for intrusion detection.

V. Conclusion

As a result, the UNSW-NB15 dataset's optimal features were chosen using the VFC algorithm for feature selection. Once an assault has been detected, the proposed Ec-k fold Cross verified Artificial neural network weighted Random Forest classification is used to make the proper classification. For the UNSW-NB15 Dataset, the suggested system VFS-EC achieved the greatest accuracy rate of 0.9987, demonstrating that the proposed method outperforms existing systems for classifying and detecting diverse threats. As a result, the system is effectively repaired in a short period of time.

References

[1] H. Alqahtani, I. H. Sarker, A. Kalim, S. M. M. Hossain, S. Ikhlaq, and S. Hossain, "Cyber intrusion detection using machine learning classification techniques," in International Conference on Computing Science, Communication and Security, 2020, pp. 121-131..

[2] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," Transactions on Emerging Telecommunications Technologies, vol. 32, p. e4150, 2021.

[3] T. Saranya, S. Sridevi, C. Deisy, T. D. Chung, and M. A. Khan, "Performance analysis of machine learning algorithms in intrusion detection system: A review," Procedia Computer Science, vol. 171, pp. 1251-1260, 2020.

[4] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in Internet of Things," Journal of Network and Computer Applications, vol. 84, pp. 25-37, 2017.

[5] S. Pundir, M. Wazid, D. P. Singh, A. K. Das, J. J. Rodrigues, and Y. Park, "Intrusion detection protocols in wireless sensor networks integrated to Internet of Things deployment: Survey and future challenges," IEEE Access, vol. 8, pp. 3343-3363, 2019.

[6] E. Benkhelifa, T. Welsh, and W. Hamouda, "A critical review of practices and challenges in intrusion detection systems for IoT: Toward universal and resilient systems," IEEE Communications Surveys & Tutorials, vol. 20, pp. 3496-3509, 2018.

[7] S. K. Biswas, "Intrusion detection using machine learning: A comparison study," International Journal of pure and applied mathematics, vol. 118, pp. 101-114, 2018.

[8] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," applied sciences, vol. 9, p. 4396, 2019.

[9] J. Zhang, R. Gardner, and I. Vukotic, "Anomaly detection in wide area network meshes using two machine learning algorithms," Future Generation Computer Systems, vol. 93, pp. 418-426, 2019.

[10] Y. He, S. Nazir, B. Nie, S. Khan, and J. Zhang, "Developing an efficient deep learning-based trusted model for pervasive computing using an LSTM-based classification model," Complexity, vol. 2020, 2020.

[11] J. Ren, J. Guo, W. Qian, H. Yuan, X. Hao, and H. Jingjing, "Building an effective intrusion detection system by using hybrid data optimization based on machine learning algorithms," Security and Communication Networks, vol. 2019, 2019.

[12] N. Bindra and M. Sood, "Detecting DDoS attacks using machine learning techniques and contemporary intrusion detection dataset," Automatic Control and Computer Sciences, vol. 53, pp. 419-428, 2019.

[13] Y.-F. Hsu, Z. He, Y. Tarutani, and M. Matsuoka, "Toward an online network intrusion detection system based on ensemble learning," in 2019 IEEE 12th international conference on cloud computing (CLOUD), 2019, pp. 174-178.

[14] H. Wang, Z. Cao, and B. Hong, "A network intrusion detection system based on convolutional neural network," Journal of Intelligent & Fuzzy Systems, pp. 1-15, 2019.

[15] Y. Liu, S. Liu, and X. Zhao, "Intrusion detection algorithm based on convolutional neural network," DEStech Transactions on Engineering and Technology Research, 2017.

[16] X. Yang and Z. Hui, "Intrusion detection alarm filtering technology based on ant colony clustering algorithm," in 2015 Sixth International Conference on Intelligent Systems Design and Engineering Applications (ISDEA), 2015, pp. 470-473.

[17] C. 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.

[18] W. L. Al-Yaseen, Z. A. Othman, and M. Z. A. Nazri, "Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system," Expert Systems with Applications, vol. 67, pp. 296-303, 2017.

[19] H. Ji, D. Kim, D. Shin, and D. Shin, "A Study on comparison of KDD CUP 99 and NSL-KDD using artificial neural network," in Advances in computer science and ubiquitous computing, ed: Springer, 2017, pp. 452-457.

[20] B. Selvakumar and K. Muneeswaran, "Firefly algorithm based feature selection for network intrusion detection," Computers & Security, vol. 81, pp. 148-155, 2019.

[21] E. Min, J. Long, Q. Liu, J. Cui, and W. Chen, "TR-IDS: Anomaly-based intrusion detection through text-convolutional neural network and random forest," Security and Communication Networks, vol. 2018, 2018.

[22] L. Xiao, X. Wan, X. Lu, Y. Zhang, and D. Wu, "IoT security techniques based on machine learning: How do IoT devices use AI to enhance security?," IEEE Signal Processing Magazine, vol. 35, pp. 41-49, 2018.

[23] B. Riyaz and S. Ganapathy, "A deep learning approach for effective intrusion detection in wireless networks using CNN," Soft Computing, pp. 1-14, 2020.

[24] J. Kim, J. Kim, H. Kim, M. Shim, and E. Choi, "CNN-Based Network Intrusion Detection against Denial-of-Service Attacks," Electronics, vol. 9, p. 916, 2020.

[25] B. Susilo and R. F. Sari, "Intrusion Detection in IoT Networks Using Deep Learning Algorithm," Information, vol. 11, p. 279, 2020.

[26] J. Jeon, J. H. Park, and Y.-S. Jeong, "Dynamic Analysis for IoT Malware Detection with Convolution Neural Network model," IEEE Access, 2020.

[27] M. Almiani, A. AbuGhazleh, A. Al-Rahayfeh, S. Atiewi, and A. Razaque, "Deep recurrent neural network for IoT intrusion detection system," Simulation Modelling Practice and Theory, vol. 101, p. 102031, 2020.

[28] D. Zheng, Z. Hong, N. Wang, and P. Chen, "An improved LDA-based ELM classification for intrusion detection algorithm in IoT application," Sensors, vol. 20, p. 1706, 2020.

[29] Q. Tian, D. Han, M.-Y. Hsieh, K.-C. Li, and A. Castiglione, "A two-stage intrusion detection approach for software-defined IoT networks," Soft Computing, pp. 1-17, 2021.

[30] Z. Wang, Z. Xu, D. He, and S. Chan, "Deep logarithmic neural network for Internet intrusion detection," Soft Computing, vol. 25, pp. 10129-10152, 2021.

[31] D. Gaifulina and I. Kotenko, "Selection of Deep Neural Network Models for IoT Anomaly Detection Experiments," in 2021 29th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), 2021, pp. 260-265.

[32] B. Roy and H. Cheung. A deep learning approach for intrusion detection in internet of things using bi-directional long short-term memory recurrent neural network. In 2018 28th International Telecommunication Networks and Applications Conference (ITNAC), pages 1–6. IEEE, 2018.

[33] H. C. Altunay, Z. Albayrak, A. N. Ozalp, and M. C¸ akmak. Analysis of anomaly ¨ detection approaches performed through deep learning methods in scada systems. In 2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), pages 1–6. IEEE, 2021.

[34] H. Dhillon and A. Haque. Towards network traffic monitoring using deep transfer learning. In 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pages 1089–1096. IEEE, 2020.

[35] S. Mohammadi, V. Desai, and H. Karimipour. Multivariate mutual information based feature selection for cyber intrusion detection. In 2018 IEEE electrical power and energy Conference (EPEC), pages 1–6. IEEE, 2018.

[36] E. Nyakundi. Using support vector machines in anomaly intrusion detection. Master's thesis, University of Guelph, 2015.

[37] M. Xie, J. Hu, and J. Slay. Evaluating host-based anomaly detection systems: Application of the one-class svm algorithm to adfa-ld. In 2014 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), pages 978–982. IEEE, 2014.

[38] M. Almseidin, M. Alzubi, S. Kovacs, and M. Alkasassbeh. Evaluation of machine learning algorithms for intrusion detection system. In 2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY), pages 277–282. IEEE, 2017.

[39] A. Bhuvaneshwari , Dr. S. Britto Ramesh Kumar 2, "An Enhancing Canny Edge Detection Using SUSAN Filter Based On Deep Learning Algorithm For Improved Image Edge Detection" Webology (ISSN: 1735-188X) Volume 18, Number 6, 2021.

[40] A. Bhuvaneshwari and S. Britto Ramesh Kumar, "AN EFFICIENT BRITWARI TECHNIQUE TO ENHANCE CANNY EDGE DETECTION ALGORITHM USING

DEEP LEARNING" ICTACT JOURNAL ON SOFT COMPUTING, APRIL 2022, VOLUME: 12, ISSUE: 03.

J. Vimal Rosy has completed her Masters in Computer Science, Masters in Philosophy Computer Science, and is currently pursuing her Ph.D. in Computer Science in the Field of Cloud Security. She currently serves as an Head & Assistant Professor in the Department of Computer Science, Soka Ikeda College of Arts and Science for Women, Chennai, Tamil Nadu, India.

Dr. S. Britto Ramesh Kumar is an Assistant Professor of Computer Science at St. Joseph's College (Autonomous), Tiruchirappalli. His research interests include software architecture, wireless and mobile technologies, information security and Web Services. He has published many journal articles and book chapters on the topics of Mobile payment and Data structure and algorithms. His work has been published in the International journals and conference proceedings, like JNIT, IJIPM, IEEE, ACM, Springer and Journal of Algorithms and Computational Technology, UK. He was awarded as the best researcher for the year 2008 in Bishop Heber College, Tiruchirappalli. He has completed a minor research project. He has visited countries like China, South Korea and Singapore.