Intelligent Automation & Soft Computing
DOI:10.32604/iasc.2022.022211
Article

Tech Science Press

Optimized Fuzzy Enabled Semi-Supervised Intrusion Detection System for Attack Prediction

Gautham Praveen Ramalingam[1], R. Arockia Xavier Annie[1] and Shobana Gopalakrishnan[2,*]

[1]Department of Computer Science and Engineering, CEG, Anna University, Chennai, 600025, India
[2]Department of Information Technology, Loyola ICAM College of Engineering and Technology, Chennai, 600034, India
*Corresponding Author: Shobana Gopalakrishnan. Email: gopal.shobana@gmail.com
Received: 31 July 2021; Accepted: 28 September 2021

Abstract: Detection of intrusion plays an important part in data protection. Intruders will carry out attacks from a compromised user account without being identified. The key technology is the effective detection of sundry threats inside the network. However, process automation is experiencing expanded use of information communication systems, due to high versatility of interoperability and ease off 34 administration. Traditional knowledge technology intrusion detection systems are not completely tailored to process automation. The combined use of fuzziness-based and RNN-IDS is therefore highly suited to high-precision classification, and its efficiency is better compared to that of conventional machine learning approaches. This model increases the accuracy of intrusion detection using Machine Learning Methodologies and fuzziness has been used to identify various categories of hazards, and a machine learning approach has been used to prevent intrusions. As a result, the hypothesis of security breaches is often observed by tracking system audit reports for suspicious trends of system use, and access controls for granting or limiting the degree of access to the network are often established as the result of an improvement in the detection accuracy of intrusions which is extremely effective.

Keywords: Intrusion detection system (IDS); recurrent neural network (RNN); security attacks; fuzzy logic; deep learning

1 Introduction

In the past couple of years, Intrusion Detection Systems (IDS) have designed unique and proprietary communication networks to determine the monitoring and control functions are isolated from public networks and to use compassable methods to mask network attack payload and avoid detection. Many supervised and unsupervised learning contributions in the field of machine learning and pattern recognition will improve the performance of intrusion detection systems. Today, such networks are heavily interconnected with conventional business systems and encapsulate modern control protocols in conventional networking protocols such as the Transmission Control Protocol (TCP)/Internet Protocol (IP), which reflect a broad variety of Inter-Connected systems (IC's) that are linked to the physical world, since IC's are widespread.

Yin et al. [1], a major scientific advance in the field of information technology will consider an attack that may be a continuous invasion or an intrusion that has already existed. They built intervention to determine whether or not the network traffic operation is normal. Chen et al. [2], suggested a system for detecting unknown attacks. The special data set is used to assess the functionality of all classification techniques for the identification of unknown attacks with the benefit of cumulative records (training examples) of known attacks. Ashfaq et al. [3], the key research is conducted on a single hidden layer feed-forward neural network (SLFN) [4–6], to have a fuzzy membership vector function in [7], and sample categorization function (low, medium, and strong fuzzy values) on unlabeled samples has been done using a fuzzy function [8]. After incorporation into the initial training collection of each division, the classifier shall be retrained separately for reducing the classification error rate. The clustering learning algorithm is looking for correlations between clustering instances and is useful in identification of attacks. Chen et al. [9], implemented an integral of the Laplacian smooth twin function on the support vector machine to make these squares differentiable and is further optimized to solve the system effectively, which converges globally and quadratic. Proceedings that belong to a certain cluster are assumed to have evolutionary characteristics indicated in [10], or properties such that the same class may allow unsupervised learning and semi-supervised learning techniques to offer a higher alternative to extending the reliability of the IDS.

Purpose of this work is to resolve the problem of intrusion detection by offering a statistical mechanism for intrusion detection systems, based on the premise that security breaches are often identified by monitoring computer audit logs for abnormal network patterns of system use. This could be achieved by fuzziness on a semi-supervised learning approach using unlabeled samples using a supervised learning algorithm to enhance the efficiency of the classification in IDS. Intrusion identification is typically analogous to a classification problem, such as a binary or multi-class classification problem, i.e, whether network traffic activity is normal or anomalous. Through concept, the classification of artificial neural network where the connections between units form a directed process that enables the complex temporal operation of arbitrary input sequences and thus the hidden-node parameters in neural networks with random weights are chosen randomly and appropriately to determine the correlation between fuzziness created by the classification model on a bunch of samples by utilizing the concept of Ashfaq et al. [3]. Alfya et al. [11] proposed an anomaly-based IDS countermeasure based on fuzzy classification and greedy attributes. This would aid in decreasing the dimensionality of the dataset and increasing computing efficiency by lowering the false-positive rate. The Combined Theory of Deep Learning and Statistical methods, which is used to classify both the attack category and the results, achieves greater efficiency and recognition efficiency with an occasional false positive rate, especially in the Multi-Class Classification Task. This framework will effectively boost both the accuracy of network security and the ability to recognize the type of intruder, which is why deep learning principles help to make correct intrusion decisions and help hunt for an intruder and increase performance by incorporating semi-supervised learning into the model and optimizing intrusive behavior and accuracy.

In earlier studies, a variety of approaches are based on conventional machine learning, including Support Vector Machine (SVM) [12,13], K-Nearest Neighbor (KNN) [14], Artificial Neural Network (ANN) [15,16], Random Forest (RF) [17,18] and others [19–21], which have been suggested and achieved success in the intrusion detection system.

Ashfaq et al. [3], introduced an alternative Semi Supervised Learning algorithm named Semi Supervised Learning using privileged knowledge. Their solution increases the classifier's precision by leveraging details on residual distribution geometry found in unlabeled data and privileged experience to maximize learning performance. Experimental studies have demonstrated that tests between low and high levels of fuzziness plays a significant role in increasing the accuracy of the IDS by preliminary work by categorizing the data according to the quantity of fuzziness.

Denning et al. [22], proposed a Deep Learning-Based Approach on an NSL-KDD benchmark dataset consisting of sensitive information which is used on a network intrusion detection system [20]. On the

NSL-KDD data collection, the repeating neural networks have built up a directional loop that can store the historic information and apply it to the current output, *i.e.*, the conventional neural feed-forward networks [6]. Previous data is typically compared to this series production, and nodes between neural network hidden layers are no longer connected; instead, relationships are necessary. Not only the output of the input nodes, but also the output of the input nodes, is considered for processing. The final secret layer is on the inside of the hidden layer which specifies a variety of techniques such as Fuzziness Algorithm, Neural Network with Random Weights, Divide and Conquer Strategy, Numerical Form, Forward Propagation, Weight Update and Classification and Advanced Deep Learning Neural Network Model for Intrusion Detection Systems. Javaid et al. [23], his work has provided IDS through similar Deep Learning method that has performed well in detection accuracy. Similarly, Liu et al. [24], has implemented a robust regression approach to the learning of discriminatory character-based learning descriptors, which is the most rational way of getting potential features from the dataset and the other handcrafted and learned functions used for classifying the threats.

2 System Design

The proposed methodology to intrusion detection shown in Fig. 1: The focus is on a semi-supervised learning approach and a fuzzy-based framework by using unlabeled samples backed by supervised learning methods to improve the efficiency of the IDS classifier and a deep learning [25], approach to intrusion detection using recurrent neural networks. It is designed to detect an invasion, and also to detect an ongoing invasion or invasion that has already occurred. It is generally used for the analysis of Signature based attacks and Anomaly based attacks. Here, the machine learning techniques are used to detect a variety of threats which can allow the network administrator to take appropriate measures to prevent intrusions. This can be achieved by a hybrid fuzziness method based on semi-supervised learning and a machine-based classification approach implemented to the NSL-KDD data set through a sophisticated connecting architecture. These concepts are applied in such a way as to accurately measure time as a whole, based on the presumption that potential threats will be identified by reviewing system reports for unusual patterns of system use thereby improving the accuracy of the intrusion and if an attack is detected or suspicious behavior is detected, the warning will be sent to the administrator. This invasion and maybe even Denial of Service (DoS), User to Root (U2R), Probe (Probing) and Root to Local (R2L) are some forms of attacks that will be listed. It was therefore also planned to examine the propagation of traffic over the entire subnet and to match the traffic transmitted over subnets to the library of documented attacks.

The work focuses on the Intrusion Detection methodology that utilizes the Fuzzy-based Semi-Supervised Learning and Deep Learning Principles. This can be done by sniffing out network packets. Network packets are also the Transmission Control Protocol (TCP), the User Datagram Protocol (UDP) or the Internet Control Message Protocol (ICMP). For this reason, the NSL-KDD dataset [26] has been used. It consists of the following network packet information: Length, Protocol type, Service type, Flag, Source Byte, Destination Byte, Server Count, Source Error Rate, Receiver Error Rate, Server Error Rate, Error Fragment and so on. These information systems can use a broad range of technology, such as distributed data management systems, encryption and security mechanisms, access control and web services. Tavallaee et al. [27], have been statistically defined and identified a variety of defects inside the original KDDCUP′99 dataset have been statistically described and established which adversely affect the effectiveness of the analyzed mechanism and have incorrect anomaly detection schemes. They proposed better data collection in the NSL-KDD data set to solve these concerns and provided a more realistic comparative system using various classifier models. There are various types of attacks in the enhanced dataset, a rank mark indicating an instance's position is either normal or an attack. The NSL-KDD dataset [25–27] comprises 41 features and 1 class label for each traffic record, including some essential features (from 1 to 10), content features (from 11 to 22) and traffic features (from 23 to 41). Following the attacks, they were identified as Denial of Service (DoS) [27], User to Root (U2R), Remote to Local

(R2L), Probing (PROBE). Hernandez–Pereira [28], examined the NSL-KDD dataset intrusion detection system using deep-neural networks. The system is structured to classify low-level spatial features of network traffic using deep convolutional neural networks (CNN's) and accompanied by memory followed by long-term memory (LSTM) networks to define high-level temporal features. These computational approaches rely on the assumption where the models can derive input from dynamically developed traffic features. Applications in the fields of computer vision and language processing have demonstrated, however, that deep neural networks are most advantageous in their ability to understand features explicitly from details. This research method used deep neural networks to support explicitly designed features without enabling the full use of deep neural networks.

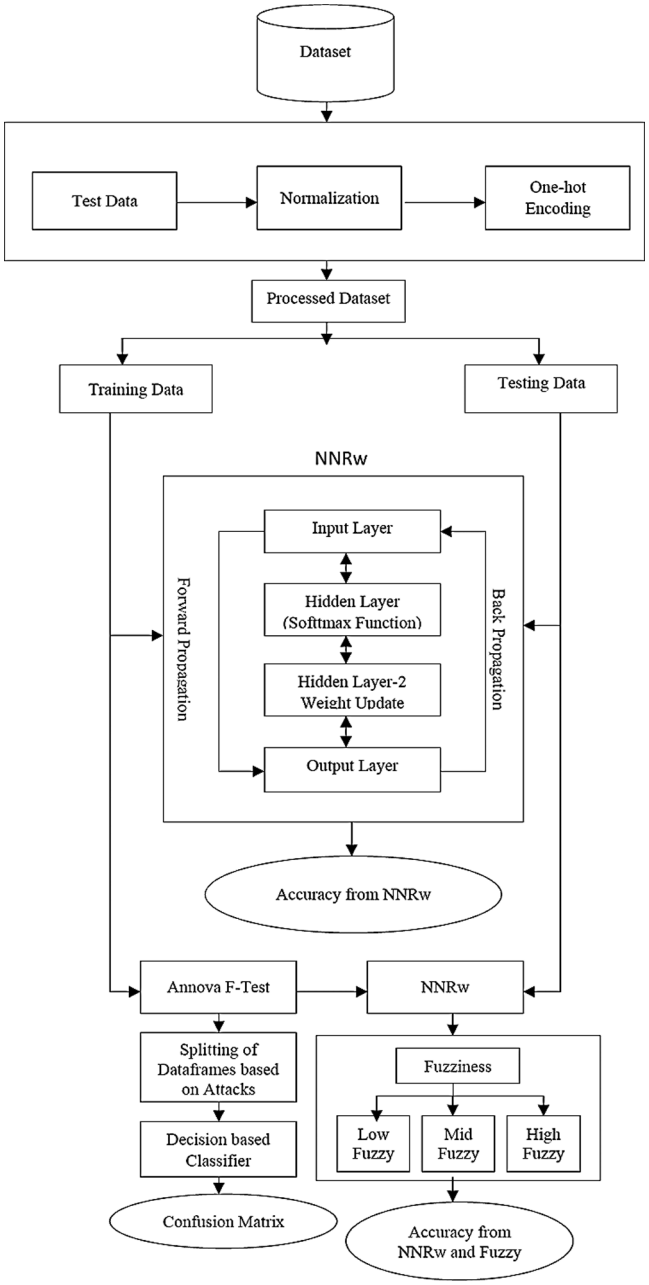

Figure 1: Architecture diagram shows the embedded semi-supervised and fuzzy based learning model focused on improved IDS classification

2.1 Data Preprocessing

Data preprocessing is a raw data preparation method suitable for a machine learning model. It includes, (i) Data Normalization and (ii) One-hot encoding.

2.1.1 Data Normalization

Dimensions are a crucial consideration when utilizing deep learning. Data normalization is used to solve this problem. Because the NSL-KDD dataset has 41 dimensions with widely varying values. As a result, the min–max normalization approach is used to minimize the various dimension scales. It is scaled between [0, 1] using a linear modification of the original data. This may be accomplished using the following equation,

$$Z_{ai}^{\sim} = (Z_{ai} - min_{ai})/(max_{ai} - min_{ai}) \tag{1}$$

where min_{ai} and max_{ai}, represent the minimum and maximum values of the numeric feature Z_{ai} respectively, and $Z_{ai}^{\sim}$ indicates that the normalized feature value is between (0,1).

2.1.2 One-Hot Encoding

One-hot encoding of categorical characteristics is a simple and effective encoding method. It is capable of converting the value of each category characteristic into a binary vector with just one element with a value of 1 and all other elements being zero. An element with a value of 1 indicates the presence of potential values for the category characteristic.

In the NSL-KDD dataset [26], the protocol feature has three distinct values, the flag feature has 11 distinct values, and the operating feature has 66 distinct values. Andresini et al. [29], adopted the encoding notion to present a multistage model with a neural network convolution layer and two stacked fully linked layers. To reconstruct the data anew, two Encoding notions were learned independently based on normal and attack flows. As NNRw is therefore unable to process symbolic or discrete data, different techniques may be used to transform symbolic data into continuous data without impacting output of the system. Some features like land, logged_in, is_host_login and is_guest_login have values of 0 or 1, so we can treat these features in the same way as continuous features. Other elements such as protocol, operation and flag have more than two distinct values.

Ashfaq et al. [3], we have obtained a better approach to handling all NSL-KDD dataset attack types and its categories are seen in Tab. 1.

Table 1: Attack types and its categories

Attack	Characteristics
Denial of service (DoS)	Back, Ping of death, Neptune, Smurf, Land and Teardrop.
User to root (U2R)	Perl, Buffer overflow, Load module and Rootkit.
Remote to local (R2L)	FTP write, Guess password, IMAP, Multi-HOP, Phf, SPY, Wareclient and Warezmaster.
Probing (PROBE)	IP-Sweep, NMAP, Port sweep and Satan.

Many researchers found various approaches to Indicator (or Dummy Variables) and the possibility of using symbolic functions. In our experiment, we are using the scheme suggested by Neter et al. [30], to deal with related functions. The flag and the repair characteristics have a variety of symbolic qualities. The scheme proposed in [31], would increase the dimensionality of the dataset and thus, accepted domain awareness, the clustering technique proposed by Hernandez-Pereira et al. [28], would also be used for

group-related categories with different symbolic characteristics. Flag and Service functions are then more grouped together to scale down the dimensions before transforming these forms into indicator variables. The classification of the Symbolic features and its distinct categories will be as follows, (i) Symbolic features of the Protocol have 3 different types and they are tcp, udp, icmp. (ii) Symbolic features of the Service have 66 different types and they are smtp, ntp_u, shell, kshell, imap4,… (iii) Symbolic features of the Flag have 11 different types and they are RSTR, S3, SF, RSTO, SH, OTH, S2, RSTOS0, S1, S0, REJ. (iv) Symbolic features of the Land have 2 different types and they are 0 and 1. (v) Symbolic features of the logged_in have 2 different types and they are 0 and 1. (vi) Symbolic features of the is_host_login have 2 different types and they are 0 and 1. (vi) Symbolic features of the is_guest_login have 2 different types and they are 0 and 1.

According to Hernndez-Pereira et al. [28], the flag feature determines the relationship condition and thus the divisions of these elements are further grouped into different categories. The values of the flag and service features grouped in the dataset are defined as follows:

- The flag cluster FG1 and its groups are S0, REG.
- The flag cluster FG2 and its groups are S1, SF, OTH.
- The flag cluster FG3 and its groups are S2, RSTO.
- The flag cluster FG4 and its groups are S3, RSTR.
- The flag cluster FG5 and its groups are SH, RSTOSO.
- The flag cluster FG6 and its groups are SHR, RSTRH.
- The service cluster SG1 and its groups are telnet, ssh, etc.,
- The service cluster SG2 and its groups are ftp, tftp, etc.,
- The service cluster SG3 and its groups are smtp, imap4, etc.,
- The service cluster SG4 and its groups are http, etc.,
- The service cluster SG5 and its groups are svstat, netstat, etc.,
- The service cluster SG6 and its groups are host name, domain, etc.,
- The service cluster SG7 and its groups are eco_i, tim_i, ecr_i, urp_i, etc.,
- The remaining services will be listed in SG8.

2.2 Softmax Function

Softmax, or generalized exponential, can be the generalization of a logistic function in which the K-dimensional vector converts the K vector of the real values into the K vector of the actual values of that sum to 1. Using this tool, the modified numeric attributes of the NSL KDD dataset are translated to real value K-dimensional vectors which are used as output prediction to a categorical probability variable. The methodology for the application of the softmax function is as follows:

i) $x_{t,}$ x is the input at time step t.
ii) The nonlinearity function Softmax Function is applied.
iii) A K-dimensional vector with actual values between 0 and 1 will be obtained.

2.3 Fuzziness Function

The word fuzziness refers to the complexity of the border between two linguistic concepts, which is focused on the linear model of the fuzzy set. It is a measure of the probability of an occurrence involving a fuzzy occurrence and suggests the use of uncertainty in mathematical theory to explain the ambiguity associated with a fuzzy occurrence. The algorithm steps to perform fuzzification is as follows:

i) Let 'x' be the NSL-KDD dataset as an input of the system.
ii) Set of fuzzy rules, such as union, intersection and negation operators, has been applied.
iii) The input variables use the input values of the extension principal function to fuse their array to determine the level of membership function of that input value to all the fuzzy variable sets, and each law applies to some degree to the output variables, and the entirety of that input will determine the output of the system.
iv) For each input element, fuzzification is done by extracting the degree of membership of all sets and by applying the rules when they are evaluated.
v) The activation value is paired with the related Fuzzy Set using the min operator, which will serve as a threshold for the degree of membership of the Fuzzy Set.
vi) Execution of the Fuzzy Rules Output Distribution Sets of all output variables will now include the inputs from each rule.
vii) Computing fuzzy sets and variables and gaining information about the inference phase execution.

Luca et al. [32] also developed the probability estimation of a fuzzy occurrence and proposed the use of entropy in information theory to interpret the complexity of a fuzzy event. They considered fuzziness to be a form of complexity and also described a quantitative definition of fuzziness with non-probabilistic probability proportional to Shannon's entropy values. They also suggested three attributes where the fuzziness could be retained. These properties reflect that the level of fuzziness approaches its limit when the degree of membership of the group of each and every element is equal and its limit when each object either belongs to a fuzzy system or not at all. In this analysis, fuzziness is known to be a form of cognitive complexity resulting from the transformation of uncertainty from one linguistic element to another, where linguistic terminology is a fuzzy set defined in a certain universe of discourse. Based on the fuzziness value, the collections are divided into three groups: Low-fuzziness group, Mid-fuzziness group and High-fuzziness group.

Bridges et al. [33] had developed a system to detect both irregularities and abuse of the network by incorporating genetic algorithms and Fuzzy data mining technologies. Artificial intelligence approaches are now becoming the most popular in terms of their potential to adapt and improve, making them more reliable and efficient in addressing a wide variety of unpredictable problems using the most important network functionality of this approach and using the Genetic Algorithm to find the most appropriate fuzzy function parameters.

2.4 Neural Network with Random Weights (NNRw)

Artificial neural networks are trained by a probabilistic optimization technique known as stochastic gradient descent. It uses randomness to find a strong enough range of weights for the particular mapping function from inputs to outputs in the data that is being learned which has been proposed through a parallel learning structure and parameters. During learning, any node that contributes to the highest reduction of error throughout the system will be removed from the hidden node and added to the current network. As a result, the precision or accuracy of the IDS will be calculated.

De Luca et al. [32], believed that fuzziness was a concept of sophistication and also established a non-probabilistic quantitative definition of fuzziness comparable to the information entropy of Shannon. They also proposed three features where the fuzziness could be preserved. These properties demonstrate that the degree of fuzziness should be as high as possible when the membership function of each entity is equal and its minimum if each element belongs to a fuzzy set or not at all. Random Vector Functional Link network, which integrates random hidden layer weights and biases, and it is a direct link between the input layer and the output layer. The membership vector function of each unlabeled sample we obtain during this step is further used to obtain F(V) fuzziness. Based on the fuzziness value, the samples are

categorized into three categories, *i.e*., the low fuzziness group FGLow, the mid fuzziness group FGMid and the high fuzziness group FGHigh, respectively, and the samples are taken from the FGLow and FGHigh fuzziness groups.

Since, NNRw: From NSL-KDD dataset, $X = \{(x_i, t_i) \mid x_i \in R\ n, t_i \in R\ m, i = 1,\ldots, N\}$ and a hidden node output function g (w, b, x), and number of hidden nodes L. The Methodology for NNRw is as follows:

i) Input parameters w_i and b_i are randomly chosen where I = 1…L.
ii) Compute the matrices of neural network throughput H.
iii) Calculate the generated output weight β.

2.5 Anova F-Test

Anova F-Test is a mathematical tool used to test the discrepancy between two or more groups. It is an optimization in which all Ts research samples were split into three groups according to the degree of fuzziness and in the initial training set of Tr, the group with the highest precision was added. The readjustments were carried out with the most recent instruction set Tr′. Their proposed approach is considered to be a Semi-Supervised Learning method in which the learning process requires some samples with unknown labelling with low fuzziness.

In this, Tr: Labeled dataset ($x_i, y_i \mid 1 \le i \le N$), Ur: Unlabeled dataset (Ui, $\mid 1 \le i \le U$), Ts: Test dataset (ti, yi $\mid 1 \le i \le K$) and the Classifier of Neural Network with Random Weights with the hidden node output function [34] of $g(z) = 1/(1+ e^{-z})$ used to calculate the accuracy using the hidden nodes. The methodology for the Anova F-Test is shown below:

i) F′ = Classifier of NNRw (Tr).
ii) Generate the F′(U).
iii) Get membership vector V for each unknown label from F′(U).
iv) Determine the Fuzziness value from each U sample.
v) Sample categorization for FGLow, FGMid, and FGHigh.
vi) Tr new = Tr + (FGLow + FGHigh).
vii) F′ = Classifier of NNRw (Tr new).
viii) Generate F′(Ts).

3 Testing

Testing, model contains two components, Forward Propagation and Back-propagation. Forward Propagation is responsible for the determination of target value, and Back Propagation is responsible for transferring residuals obtained to modify weights that are not inherently separate from standard neural network training. In this, the forward distribution of input data would be fed across the network in the forward direction. Input data were accepted from each hidden layer, processed according to the activation function, passed towards the next layer, and back-propagation is commonly used to train neural networks. When the neural network is initialized, its individual members, called neurons, are weighted. The inputs are loaded, passed into the neurons network, and the network generates output for each, provided the initial weights. As a result, the consistency of the IDS is achieved.

3.1 Forward Propagation

Schmidt et al. [35], who have previously investigated the effect of the genetic algorithm initialization on the effectiveness of single-layer neural feed-forward networks for generalization. Experimentally, it has been shown that a single layer feed-forward neural network can improve efficiency by selecting the random

weights associated with the input layer and by evaluating the weights of the output layer and Blum et al. [34], suggested a method to improve bias in estimating the generative model from the classification model by measuring the generative model from both labelled and unlabeled data by modelling the missing labels from a hidden variable within the mixed theoretical model. The research thus requires the non-iterative processing of neural networks by randomness. Amiri et al. [36], concluded that the weights of the output layer are substantially more significant than the weights contained in the convolution neurons of the single layer feed-forward neural network. W_{hh} is the hidden-to-hidden weight matrix, W_{yh} is the hidden-to-output weight matrix, and bh and by are the bias vectors. This sequence of predictions could be made. The methodology is as explained below:

i) From the initial information as input X, rendered to propagate into weight parameters, it is important to update each hidden layer (W_{hx}, W_{hh}, W_{yh}) present in and produce the intermediate value output of every node.
ii) Calculate the Activation function to the Intermediate output value for the duration of the vector.
iii) Sigmoid Function needs to be implemented for regularization.
iv) Weight improvement of bias in the regularization process has been carried out.
v) Compute the Softmax function, each and every node in and out of the hidden layers.

3.2 Weight-Update Algorithm

The weight update algorithm in [37], applicable to a broad range of difficulties in learning and optimization. The process will have a probability distribution of weight over a given set, which is modified recursively by the law of multiplication. The study of these algorithms focuses on quantifying the change in the potential exponential equation. The mechanism of changing multiplicative weights is the algorithmic technique most commonly used for decision-making and prediction.

Yam et al. [37], had a method to initialize neural network weights prior to back-propagation training. Similarly, a neural network with a random weight randomly initializes convolutional layers and is fitted using a generalized-inverse layer. From this, it can be shown that many proposals have been proposed for randomizing the hidden layer throughout the Neural Network, including the Random Vector Functional Link (RVFL) network, which incorporates arbitrary hidden layer weights and biases, and the possible correlation between the input layer and the output layer. For a weight update algorithm, a single training pair (x_i, y_i) is defined as $f(\theta) = L(y_i:\hat{y}_i)$, Where L is a distance function that calculates the difference between predictions and unique labels y_i. Let η be the training rate of the system and K is the number of existing implementations of the system. The methodology to perform weight update is as follows:

i) for i from k down to 1 do
ii) Measure the cross-entropy between the output value and the input value:
iii) Compute the partial derivative with respect to θ_i:
iv) The weight adjustments can be done for every node in the neuron:
v) end for

3.3 Backpropagation

Backpropagation, supervised learning of artificial neural networks, which measures the gradient of loss of feature in relation to the weights of a single input network–the output of a single training pair, and calculates the discrepancy between the y_i reductions and the individual points. The steps to perform backpropagation are as follows:

i) Initiate the link with weights to small random values.
ii) The test activation function of the sequence and the related output source T to the network has been fixed.
iii) For each neuron 'i' determines the output of the neuron by exponential function for each layer from the input to the output layer.
iv) Obtain output values.
v) Calculate the estimation error in the backward order from output to input layer for each node in each layer.
vi) For every node in the neuron, weight changes would be carried out.

3.4 NNRw Prediction

The aim of neural networks is to increase the performance of classifiers in the successful identification of invasive behavior. Zhou et al. [38], have developed a methodology for applying Semi-Supervised Learning on a directed graph, and the framework of the nodes along with the projection of the boundaries will be taken as a key point. This algorithm takes the directed graph and the label collection as input, and the function is computed using the labelled vertices to identify unmarked vertices. This function can also be used as a variable selection of the directed graph in the absence of labelled instances. Predicted values by classification will be given as input and output to Classify and Forecast Accuracy. The NNRw prediction is as follows:

i) From the set of U predicted data from the updated weights, and a set of I values from classification.
ii) Let R be the matrix of size $|U| \times |I|$ that contains all the predictions that the data have assigned to the network packets, used for identifying the Intrusion.
iii) To get the prediction on an intrusion, calculate the dot product of the two vectors.
iv) The error between the estimated prediction and the real prediction, can be calculated for the user-item pair.
v) Compute minimizing error to modify the gradient values from the current values.
vi) Compute two matrices ie., P and Q such that $P \times Q$.

4 Results and Discussions

Accuracy is the most critical element in assessing the efficiency of intrusion detection by ensuring that the information is accurate and without distortion. "Fig. 2: The accuracy of each attack namely, DoS, Probe, R2L and U2R are shown separately as Figs. 2a–2d respectively, which is present in the NSL-KDD dataset can be seen". In addition to those various parameters were also used namely, True-Positive (TP), False-Positive (FP), True-Negative (TN) and False-Negative (FN), Confusion Matrix, Accuracy, Recall, F1 Ranking. Dependent on the same benchmark, using NSL-KDD as the test range, the experimental findings suggest that for both binary and multiple classifications, the intrusion detection fuzziness model has spent more time testing, but using GPU acceleration will minimize training time in [16]. These studies have demonstrated that the system is capable of detecting both recognized and new modes of traffic, while at the same time having fair detection rates and false positives. In addition, the experimental results showed that the reliability of the system was improved after the latest attacks had been updated.

As far as, the main purpose behind our work is not only to locate the smallest mistake in classification, but also to try to identify a model that must be able to integrate new data that preserves its good generalization capabilities. We calculate the fuzziness of each unmarked sample given by the classier and try to discover its relationship to misclassification in order to enhance the accuracy of the system. Thus, experimental results indicate that samples belonging to the low and high fuzziness classes play a significant role in increasing the

accuracy of the IDSs. Based on the results, the accuracy of our proposed KDDTest algorithm was max. Precision achieved by Tavallaee et al. [27], where similar classifiers were used to achieve precision on both test datasets. Fig. 3: The contrast of results between the various attacks by using the NNRw classifier and fuzzy.

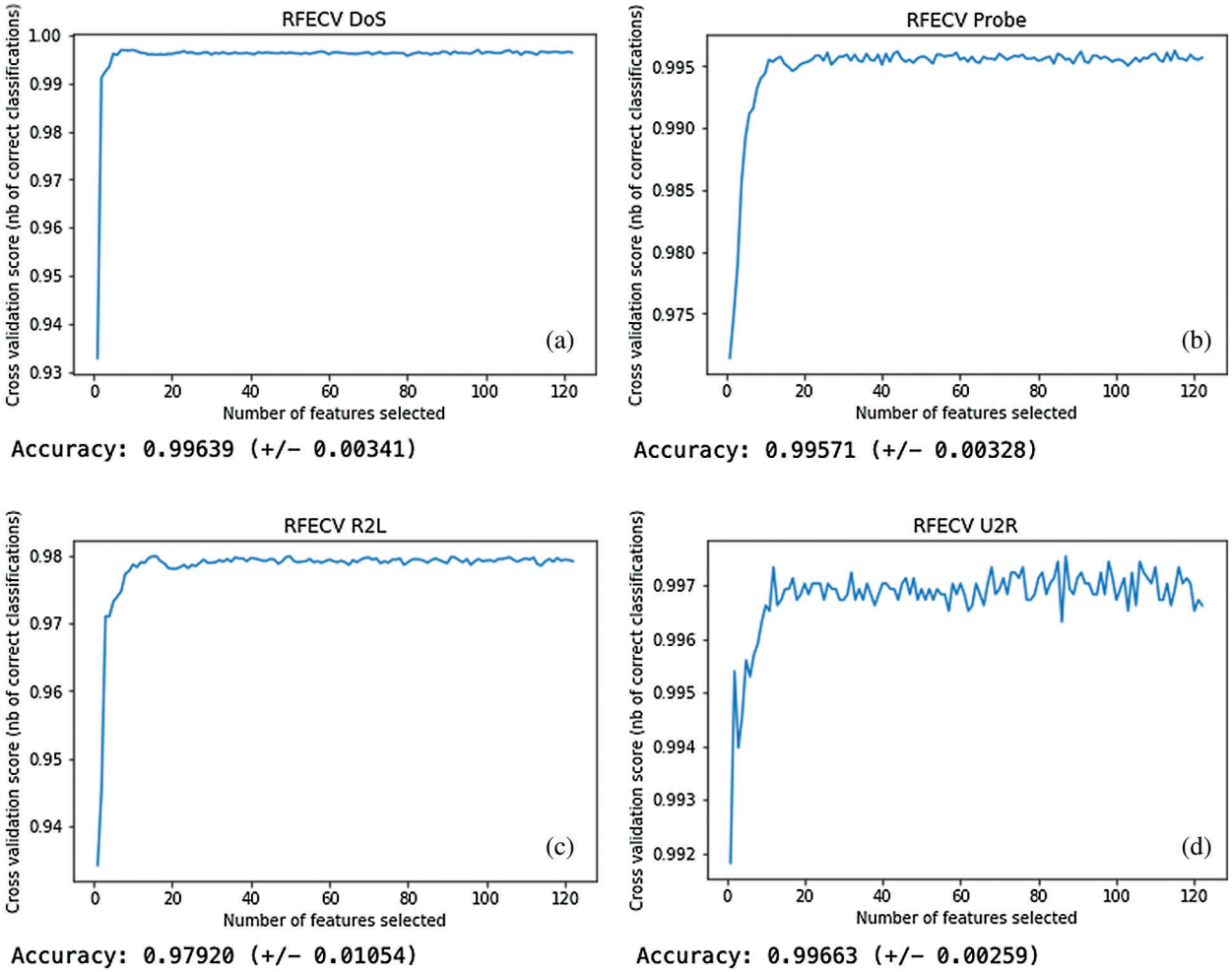

Figure 2: Shows the accuracy metrics for (a) Accuracy of DoS attack, (b) Accuracy of probe attack, (c) Accuracy of R2L attack and (d) Accuracy of U2R attack using Matplotlib, stratified K fold, cross validation score and recursive feature elimination

Furthermore, a performance comparison of various classifiers was performed in R using the same NSL-KDD dataset including testing and training data. J48 algorithm (J48), Naive Bayes Algorithm (NB), NB Tree (NBT), Random Forest (RF), Random Tree (RT), Multi-Layer Perceptron (MLP), and Support Vector Machine (SVM) were among the classifiers used. These accuracies were compared to those of Forward propagation and Backpropagation (FP & BP) and Fuzzy-based NNRw. Fig. 4: This shows the Performance comparison between the multiple classifiers and the test of our methodology and obtains the accuracy of using the NSL-KDD dataset by using various classifiers and fuzzy.

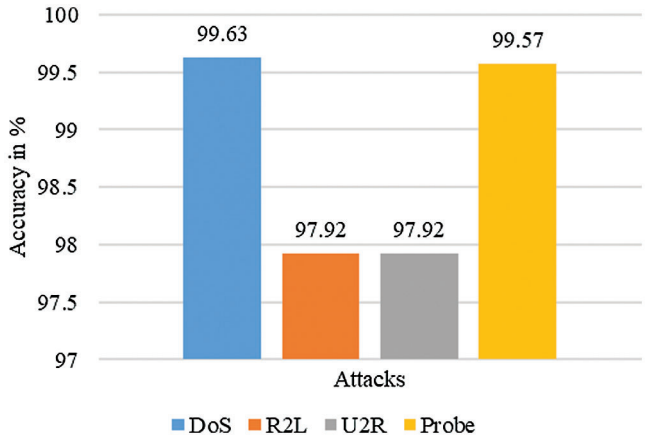

Figure 3: Shows the test of our methodology and obtains the accuracy of using the NSL-KDD dataset for different attacks by using the NNRw classifier and fuzzy

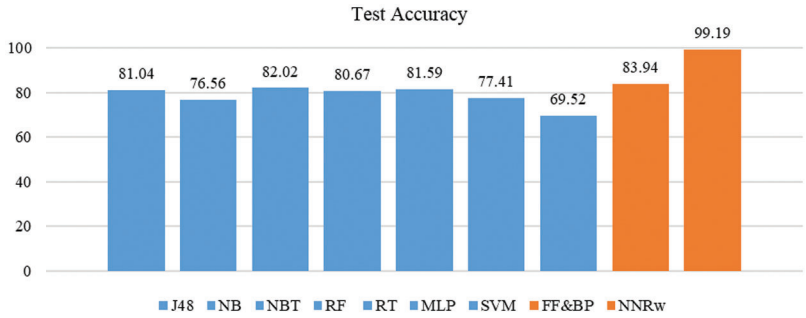

Figure 4: Shows the performance comparison between the multiple classifiers and the test of our methodology and obtains the accuracy of using the NSL-KDD dataset by using various classifiers and fuzzy

5 Conclusion

Not only does this fuzzy based architecture have a high potential for model intrusion detection, it also has a lower accuracy for both binary and multi-class classifications and with some false-positive rates. In the context of a multi-class NSL-KDD data set classification task, as opposed to a fuzzy-based Intrusion Detection System. The Fuzziness-based IDS model can quickly improve both the precision of intrusion detection and the ability to recognize the type of intrusion. The proposed IDS is an adaptive strategy that provides the opportunity to detect proven and novel attacks and to modify them on the basis of new feedback from human experts in a cost-effective manner. Research has found that all of the recommended approaches do marginally better than the conventional approach to complete retraining where the scale of the instruction range is surpassed. Experiments have demonstrated that the device is capable of recognizing both recognized and novel categories of traffic, while at the same time having reasonable identification rates and false positives.

In addition, Future research will concentrate on model performance with other characteristics, as well as other additional preprocessing and false classification approaches, and experimental results showed that the device's reliability improved after the most recent attacks were updated in the Signature database. We will continue to focus on minimizing training time using GPU acceleration in future experiments, preventing gradients from bursting, bypassing, learning of new noise in the data and vanishing and learning Long Short-Term Memory networks (LSTM) such as Bidirectional Recurrent Neural Network (Bi-RNN) performance classification in the area of intrusion detection. Larger sized network packets will be processed and large sized dataset will be with more kernels and deeper architectures can be generated by using more

number of CPU or single CPU with more processing power. Spatial and anatomical correlation of network packets is taken as prior information for processing the data. So, we have planned to develop the model for training and classifying the intruder effectively by using Generalized Adversarial Networks which will be used for the effective synthesis of potential features from the database with comparable types of threats, as well as the model's detection effectiveness with a lower false-positive rate.

Acknowledgement: The authors would like to thank the editors and reviewers for their review and recommendations.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] C. Yin, Y. Zhu, J. Fei and X. He, "Deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.

[2] C. Chen, Y. Gong and Y. Tian, "Semi-supervised learning methods for network intrusion detection," in *Proc. of IEEE Int. Conf. on Systems, Man and Cybernetics*, Singapore, pp. 2603–2608, 2008.

[3] R. A. R. Ashfaq, X. Z. Wang, J. Z. Huang, H. Abbas and Y. L. He, "Fuzziness based semi-supervised learning approach for intrusion detection system," *Information Sciences*, vol. 378, no. 1, pp. 484–497, 2017.

[4] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[5] A. Fujino, N. Ueda and K. Saito, "A hybrid generative/discriminative classifier design for semi-supervised learning," *Transaction of the Japanese Society for Artificial Intelligence* vol. 21, no. 3, 301–309, 2006.

[6] G. B. Huang, L. Chen and C. K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Transactions on Neural Networks*, vol. 17, no. 4, pp. 879–892, 2006.

[7] S. Al-Sharhan, F. Karray, W. Gueaieb and O. Basir, "Fuzzy entropy: A brief survey," *Proc. of IEEE Int. Conf. on Fuzzy Systems*, vol. 3, pp. 1135–1139, 2001.

[8] L. A. Zadeh, "Probability measures of fuzzy events," *Journal of Mathematical Analysis and Applications*, vol. 23, no. 2, pp. 421–427, 1968.

[9] W. J. Chen, Y. H. Shao and N. Hong, "Laplacian smooth twin support vector machine for semi-supervised classification," *International Journal of Machine Learning and Cybernetics*, vol. 5, no. 3, pp. 459–468, 2014.

[10] P. Bermejo, L. de la Ossa, J. A. Gamez and J. M. Puerta, "Fast wrapper feature subset selection in high dimensional datasets by means of filter re-ranking," *Journal of Knowledge Based Systems*, vol. 25, no. 1, pp. 35–44, 2012.

[11] E. S. M. Alfya and F. N. A. Obeidat, "A multicriterion fuzzy classification method with greedy attribute selection for anomaly-based intrusion detection," *The 9th Int. Conf. on Future Networks and Communications (FNC-2014)*, vol. 34, pp. 55–62, 2014.

[12] F. Kuang, W. Xu and S. Zhang, "A novel hybrid KPCA and SVM with GA model for intrusion detection," *Applied Soft Computing*, vol. 18, no. 10, pp. 178–184, 2014.

[13] R. R. Reddy, Y. Ramadevi and K. V. N. Sunitha, "Effective discriminant function for intrusion detection using SVM," in *Proc. of Int. Conf. on Advances in Computing, Communications and Informatics*, Jaipur, India, IEEE Conference, pp. 1148–1153, 2016.

[14] W. Li, P. Yi, Y. Wu, L. Pan and J. Li, "A new intrusion detection system based on KNN classification algorithm in wireless sensor network," *Journal of Electrical and Computer Engineering*, vol. 2014, no. 5, pp. 153–160, 2014.

[15] M. H. Bhuyan, D. K. Bhattacharya and J. K. Kalita, "Network anomaly detection: Methods, systems and tools," *IEEE Communication Surveys and Tutorials*, vol. 16, no. 1, pp. 303–336, 2014.

[16] B. Ingre and A. Yadav, "Performance analysis of NSL-KDD dataset using ANN," in *in proc. of Int. Conf. on Signal Processing and Communication Engineering Systems*, Guntur, India, IEEE, Conference, pp. 92–96, 2015.

[17] N. Farnaaz and M. A. Jabbar, "Random forest modeling for network intrusion detection system," *Procedia Computer Science*, vol. 89, no. 1, pp. 213–217, 2016.

[18] J. Zhang, M. Zulkernine and A. Haque, "Random-forests based network intrusion detection systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 5, pp. 649–659, 2008.

[19] G. B. Huang, Q. Y. Zhu and C. K. Siew, "Extreme learning machine: A new learning scheme of feedforward neural networks," *Proc. of IEEE Int. Joint Conf. on Neural Networks*, Budapest, Hungary, vol. 2, pp. 985–990, 2004.

[20] X. Peng, L. Wang, X. Wang and Y. Qiao, "Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice," *Computer Vision and Image Understanding*, vol. 150, no. 3, pp. 109–125, 2016.

[21] A. A. Liu, Y. T. Su, P. P. Jia, Z. Gao, T. Hao *et al.,* "Multiple/single-view human action recognition via part-induced multitask structural learning," *IEEE Transactions on Cybernetics*, vol. 45, no. 6, pp. 1194–1208, 2015.

[22] D. E. Denning, "An intrusion-detection model," *IEEE Transactions on Software Engineering*, vol. 13, no. 2, pp. 222–232, 2007.

[23] A. Javaid, Q. Niyaz, W. Sun and M. Alam, "A deep learning approach for network intrusion detection system," in *Proc. of the Conf. Bio Inspired Models of Network, Information and Computing Systems*, New York City, United States, pp. 21–26, 2016.

[24] L. Liu, L. Shao, X. Li and K. Lu, "Learning spatial-temporal representations for action recognition: A genetic programming approach," *IEEE Transactions on Cybernetics*, vol. 46, no. 1, pp. 158–170, 2016.

[25] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, no. 3, pp. 85–117, 2015.

[26] The NSL-KDD data set, University of New Brunswick (UNB). [Online]. Available: http://nsl.cs.unb.ca/NSL-KDD/.

[27] M. Tavallaee, E. Bagheri, W. Lu and A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *IEEE Symp. on Computational Intelligence for Security and Defense Applications*, Ottawa, ON, Canada Conference, IEEE, pp. 1–6, 2009.

[28] E. Hernández-Pereira, J. Surez-Romero, O. Fontenla-Romero and A. Alonso-Betanzos, "Conversion methods for symbolic features: A comparison applied to an intrusion detection problem," *Expert Systems with Applications*, vol. 36, no. 7, pp. 10612–10617, 2014.

[29] G. Andresini, A. Appice, D. N. Mauro, C. Loglisci and D. Malerba, "Multi-channel deep feature learning for intrusion detection," *IEEE Access*, vol. 8, pp. 53346–53359, 2020.

[30] J. Neter, M. H. Kutner, C. J. Nachtsheim and W. Wasserman, Applied linear statistical models. In: *WCB/McGraw-Hill Irwin*, 2005 ISBN 0071122214, 9780071122214, Length, pp. 1396. Boston, 1996.

[31] Z. Chiba, N. Abghour, K. Moussaid, A. E. Omri and M. Rida, "A novel architecture combined with optimal parameters for back propagation neural networks applied to anomaly network intrusion detection," *Computers & Security*, vol. 75, pp. 36–58, 2018.

[32] A. De Luca and S. Termini, "A definition of a non-probabilistic entropy in the setting of fuzzy sets theory," *Information and Control*, vol. 20, no. 4, pp. 301–312, 1972.

[33] S. M. Bridges and R. B. Vaughn, "Fuzzy data mining and genetic algorithms applied to intrusion detection," in *12th Annual Canadian Information Technology Security Sym.*, Canada, pp. 68–62, 2000.

[34] A. Blum and S. Chawla, "Learning from labeled and unlabeled data using graph mincuts," in *Proc. of Int. Conf. on Machine Learning*, Williams College, Williamstown, MA, USA, June 28–July 1, pp. 19–26, 2001.

[35] W. Schmidt, M. Kraaijveld and R. Duin, "Feedforward neural networks with random weights," in *Proc. of 11th Int. Association for Pattern Recognition-Inter. Conf. on Pattern Recognition*, Vol. II. Conference B: Pattern Regonition Methodology and Systems, The Hague, Netherlands, pp. 1–4, 1992.

[36] F. Amiri, M. M. Rezaei Yousefi, C. Lucas, A. Shakery and N. Yazdani, "Mutual information based feature selection for intrusion detection," *Network and Computer Application*, vol. 34, pp. 1184–1199, 2011.

[37] J. Y. Yam and T. W. S. Chow, "A weight initialization method for improving training speed in feedforward neural networks," *Neurocomputing*, vol. 30, no. 1–4, pp. 219–232, 2000.

[38] D. Zhou, J. Huang and B. Scholkopf, "Learning from labeled and unlabeled data on a directed graph," in *Proc. of Int. Conf. on Machine Learning*, Bonn Germany, August 7–11, pp. 1036–1043, 2005.