Computers, Materials & Continua
DOI:10.32604/cmc.2022.019127
Article

Tech Science Press

A Hybrid Approach for Network Intrusion Detection

Mavra Mehmood[1], Talha Javed[2], Jamel Nebhen[3], Sidra Abbas[2,*], Rabia Abid[1], Giridhar Reddy Bojja[4] and Muhammad Rizwan[1]

[1]Department of Computer Science, Kinnaird College for Women, Lahore, 54000, Pakistan
[2]ASET Labs, Islamabad, Pakistan
[3]Prince Sattam bin Abdulaziz University, College of Computer Science and Engineering, Alkharj, 11942, Saudi Arabia
[4]College of Business and Information Systems, Dakota State University, Madison, United States of America
[*]Corresponding Author: Sidra Abbas. Email: sidra.abbas708@gmail.com
Received: 03 April 2021; Accepted: 12 May 2021

Abstract: Due to the widespread use of the internet and smart devices, various attacks like intrusion, zero-day, Malware, and security breaches are a constant threat to any organization's network infrastructure. Thus, a Network Intrusion Detection System (NIDS) is required to detect attacks in network traffic. This paper proposes a new hybrid method for intrusion detection and attack categorization. The proposed approach comprises three steps to address high false and low false-negative rates for intrusion detection and attack categorization. In the first step, the dataset is preprocessed through the data transformation technique and min-max method. Secondly, the random forest recursive feature elimination method is applied to identify optimal features that positively impact the model's performance. Next, we use various Support Vector Machine (SVM) types to detect intrusion and the Adaptive Neuro-Fuzzy System (ANFIS) to categorize probe, U2R, R2U, and DDOS attacks. The validation of the proposed method is calculated through Fine Gaussian SVM (FGSVM), which is 99.3% for the binary class. Mean Square Error (MSE) is reported as 0.084964 for training data, 0.0855203 for testing, and 0.084964 to validate multiclass categorization.

Keywords: Network security; intrusion detection system; machine learning; attacks; data mining; classification; feature selection

1 Introduction

Due to deep integration between the world and the internet, the network framework always experiences various kinds of attacks. Identification of these attacks is a technical issue and currently the area of concern these days. Intrusion violates fundamental privacy conditions, e.g., confidentiality, integrity, accessibility, denial of services [1–3]. The purpose of NIDS is to identify an intrusion on networks. They detect misuse of attempts either by a legal person or by third parties [4]. Break-in security vulnerabilities, misuse of the system are attacks that the IDS can identify [5–9]. Analysis of the transmitted packet in a network and through the collection of data, IDS worked [10–12]. IDS is a classification problem to detect the behavior of data, either it is

normal or anomalous [13,14]. If network traffic is malicious, it will lie in any few of four attack categories: Denial of services, User-to-root, root-to-local, probing.

In the research related to the intruder's signature-based detection, activities that are not intrusive are also marked as anomalous behavior and generate false and false-positive results [15]. Data set preprocessing is vital to achieving the maximum success rate through the classifier. Classifier results depend on input data.

If input data is preprocessed, the results will be more accurate. However, it is a complex task and takes and more time [16]. Pre-processing reduces data complexity, so it is easy to understand the nature of data, and data analysis will be performed more accurately. Several tools and methods are available to preprocess data, including sampling, normalization, discretization data transformation, and feature extraction. Problems that cater to data preprocessing are removing noise, replacing missing values, and inconsistency in data [17].

The utilization of various data mining techniques and efficient NIDS is usually developed because they improve prediction performance, reduce computation time, and better understand data. The number of input variables is reduced to achieve desirable results to reduce computational modeling cost and increase performance. NSL KDD dataset has a large no of data that is needed to be filtered. Supervised and unsupervised are the two main feature selection techniques. The unsupervised feature selection technique works on finding the correlation between input variables [18,19] and ignores the target variable. Supervised feature selection removes the irrelevant features with the help of the target variable.

Various machine learning methods are used to identify anomalies in networks, and, as a result, it helps the network administrator take the required precautions to avoid intrusion over its network. Machine learning conventional strategies are part of shallow learning and rely on inputs. Since the data requirements for classification methods differ from one another. The two categories of machine learning methods are supervised and unsupervised. Unsupervised machine learning methods have been proven to be the most powerful in an anomaly-based intrusion detection system. Machine learning systems are composed of several computing layers and learn various data representations with several abstraction levels that fall in deep learning classes.

SVM[1] can be used for classification and works by finding a hyperplane in N-dimensional space that separates n classes. Hyperplanes are of any possible type chosen to separate two different class data points a finding a hyperplane with maximum margin results in better accuracy. Data points closer to the hyperplane influence its position and orientation and are named support vectors. In the NSL-KDD dataset, there are 42 features, so it is pretty complex to draw it. The classifier's margin is maximized with these support vectors' help, and they helped build an SVM. SVM works on the output of a linear function. If the output of a function is 1, it will classify to one class; if it is -1, then it will classify to other classes.

Fuzzy logic works on the methodology of human decision-making. As the word fuzzy refers to vague things, so it deals with vague and imprecise information. Fuzzy logic[2] is based on fuzziness instead of Boolean Logic that only results in true or false. The adaptive neuro-fuzzy inference system resembles artificial neural networks (ANN). For capturing the benefits of ANN and fuzzy in a single framework, it integrates principles of both techniques. Fuzzifier takes input and assigns a linguistic variable. If-then rules are defined in the rule base block. Rules have

[1] https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms

[2] https://www.tutorialspoint.com/fuzzy_logic/fuzzy_logic_introduction

instructions on how to deal with different variations in data. Defuzzifier will convert data into the same category at which it came in the input.

(A) *Problem Statement*

Due to deep integration between the world and the internet, the network framework always experiences various kinds of attacks. Identification of these attacks is a technical issue and currently the area of concern these days. Intrusion violates fundamental privacy conditions, e.g., confidentiality, integrity, accessibility, denial of services. The purpose of this research is to identify an intrusion on networks.

B. *Contribution*

Therefore, this paper makes the following contributions:

- Propose an Adaptive Decision Support System for Network Intrusion detection using SVM and ANFIS(hybrid) model combined with feature selection.
- Dataset is preprocessed through data transformation technique and min-max method. Secondly, Random Forest-recursive feature elimination method is applied to filter the dataset and identify optimal features that positively impact the model's accuracy.
- Next, SVM is used to decide the decision class, and the ANFIS is to detect probe, U2R, R2U, and DDOS attacks in network traffic.
- The validation of the proposed method calculated through FGSVM is 99.3% for the binary class and MSE of 0.084964 for training data, MSE of 0.0855203 for testing, and MSE of 0.084964 for validation of multiclass in NSL-KDD data.

The remaining part is composed in the following manner. Section II presents the related work. Furthermore, Section III presents the proposed methodology. Section IV presents the experiment results and analysis, and Section V presents the conclusion.

2 Related Work

No machine learning approaches have been used in prior studies. The compilation of data sets and their interpretation influence the success of the IDS. A similar data set is used for testing and training, which made difficult real-time intrusion detection. The author in [16] proposed selecting elect features from the NSLKDD data set through the correlation feature selection subset evaluation method. They then evaluate the performance by comparing their results with selected features without selecting features from the data set. According to research [17], a hybrid model in which optimal features from network transaction data are trained, and intrusion scope threshold degree can be estimated. Their proposed hybrid approach has a significant effect on algorithm time and complexity. According to [20], the author presented recurrent neural network techniques, does intrusion detection. They measure the performance of the proposed approach in both multiclass and binary class classification. They relate their suggested solution to the J48, the artificial neural network, and the random forest and support vector machines.

According to research [21], the authors analyze six classifiers on the NSL-KDD data set. They detect normal and anomalous by using, Decision tree naïve bayed, Ada boost MLP, Random forest, and SVM classifiers and implement them in python by using the spirit library. They achieved the largest accuracies from MLP 100%, and the decision tree is 98%. The authors in [22] show how to choose an appropriate classifier by testing different classifier results. Likewise, according to research [23], MLP detects the proposed recent multilayer perceptron

(MLP) network system Artificial bee colony and fuzzy clustering process, irregularities in network traffic as MLP is trained using the ABC algorithm. The Sim Cloud was used as a tool and the NSL-KDD dataset. They describe the root mean square and say the absolute error as the evaluation criteria. According to the author in research [24], a hybrid solution was suggested, incorporating a synthetic minority oversampling method, a cluster sample, and the nearest neighbor. Data is preprocessed using the Leaving One Out Process. They have used the NSL-KDD dataset, and their analysis increases the Precision of U2R and R2Las identification relative to the base paper they use. According to the author in [25], a method unit uses statistical processing to detect intrusion traffic. These systems are designed inside a plan, and these models are intended to identify rare occurrences of intrusion. According to research [26], the author developed a hypervisor detector that detects anomalies in a cloud environment at the hypervisor layer. They deploy a neuro-fuzzy system to detect anomalies in a network with a hybrid backpropagation algorithm.

According to research in [27] author discussed their work; a Wrapper Based Feature Extraction Unit (WFEU) is used to propose a Feedforward Deep Neural Network (FFDNN) wireless IDS system. UNSW-NB15 and the AWID intrusion detection datasets are used to find the effectiveness and efficiency of the WFEU-FFDNN.WFEU generates an optimal feature vector consisting of 22 attributes in the instance of the UNSW-NB15. Their approach achieved an overall accuracy of 77.16% for multiclass and 87.10% for the binary class using this input vector. Similarly, A reduced input vector of 26 attributes was generated by the WFEU in the instance of the AWID, and the experiments demonstrated that our method obtained overall accuracies of 99.6% for binary and 99.77% for the multiclass classification configurations. Authors in [28] proposed work for IDS. A wrapper feature selection algorithm to binarize a continuous pigeon-inspired optimizer is presented. Compared to the traditional way for binarizing continuous swarm intelligent algorithms, the algorithm's performance was compared to the conventional method for binarizing continuous swarm intelligent algorithms, using three popular KDDCUP99, NLS-KDD, and UNSW-NB15 datasets. According to research [29], the author discussed their work; new network architecture software-defined networking (SDN) appeared to address these challenges and provides distinctive features to cope with organizational business needs. A machine learning approach is used to design the IDS of SDN. To achieve better efficiency and accuracy, a deep learning approach is also being explored. DS as a security solution is explained to explain the SDN with its security concerns.

The author in research [30] discusses bout the security threats by the keynotes. They proposed an AlphaLogger, which is based on an android application. It typed alphabets keys by a soft touch. They use hardware sensors in smartphones to make the application more effective, with a 90.2% accuracy rate. The author in [31] represents an approach for detecting Botnet Attacks at the initial stage. The proposed scenario used machine learning algorithms for detection and compared the decision tree, PNN, SMO, and Adaboost classification technique. The author in [32], a hybrid approach has been used to defend an android application from malware attacks. This research is based on a comparison of deep learning classification techniques that studies its performance and accuracy rate.

The authors in [33] provided a deep analysis of the phishing attacks by covering machine learning, deep learning, hybrid learning, and a similar scenario-based approach to make the root cause of the attack more comprehensive. Likewise, the research [34] proposed a method for intrusion detection in the vehicular module. The study is based on the combination of CNN framework and GRU-based attention named CANintelliDS. The proposed model achieved a 10.79% intrusion

testing rate. The author in [35] presented an AI-based efficient complexity solver that has been proposed, which can help many engineers. It can also be helpful for IoT infrastructure and environment for automatic applications line Healthcare systems and smart stations for monitoring. Tab. 1 represents the previous work on many datasets for classification, clustering, increasing efficiency, cost-effectiveness, and many more.

Table 1: Relevant literature review

Ref. No.	Research topic	Description
[17]	A comprehensive approach towards data preprocessing techniques & association rules	The authors proposed a hybrid model for extracting optimal features from transaction data and compare threshold degrees.
[20]	Decision tree-based intrusion detection system for NSL-KDD dataset	They used the Neural Network approach and related their suggested solution to the J48, the artificial neural network, and the random forest and support vector machines.
[21]	Anomaly-based intrusion detection system through feature selection analysis and building efficient hybrid model	They analyzed six classifiers on the NSL-KDD data set and detect normal and anomalous using, Decision tree naïve bayed, Ada boost MLP, Random forest, and SVM classifiers.
[22]	A deep learning approach for intrusion detection using recurrent neural networks	The work showed how to choose an appropriate classifier by testing different classifier results.
[23]	Analysis of classification techniques for intrusion detection	MLP detected the proposed recent multilayer perceptron (MLP) network system Artificial bee colony and fuzzy clustering process, irregularities in network traffic.
[24]	Comparison of classification techniques applied for network intrusion detection and classification	They used the NSL-KDD dataset, and their analysis increases the Precision of U2R and R2L.
[25]	Intrusion detection for cloud computing using neural networks and artificial bee colony optimization algorithm	They proposed a method unit that uses statistical processing to detect intrusion in traffic.
[26]	A hybrid data mining approach for intrusion detection on the imbalanced NSL-KDD dataset	They developed a hypervisor detector that detects anomalies in a cloud environment at the hypervisor layer.
[27]	A deep learning method with wrapper based feature extraction for the wireless intrusion detection system	UNSW-NB15 and AWID intrusion detection datasets are used to find the effectiveness and efficiency of the WFEU-FFDNN.
[28]	A feature selection algorithm for intrusion detection based on pigeon inspired optimizer	They proposed a wrapper feature selection algorithm to binarize a continuous pigeon-inspired optimizer is proposed.

3 Proposed Framework

This section describes the proposed approach for network intrusion detection. In this work, the hybrid approach SVM with ANFIS is applied for effective detection. Fig. 1 shows the proposed approach.

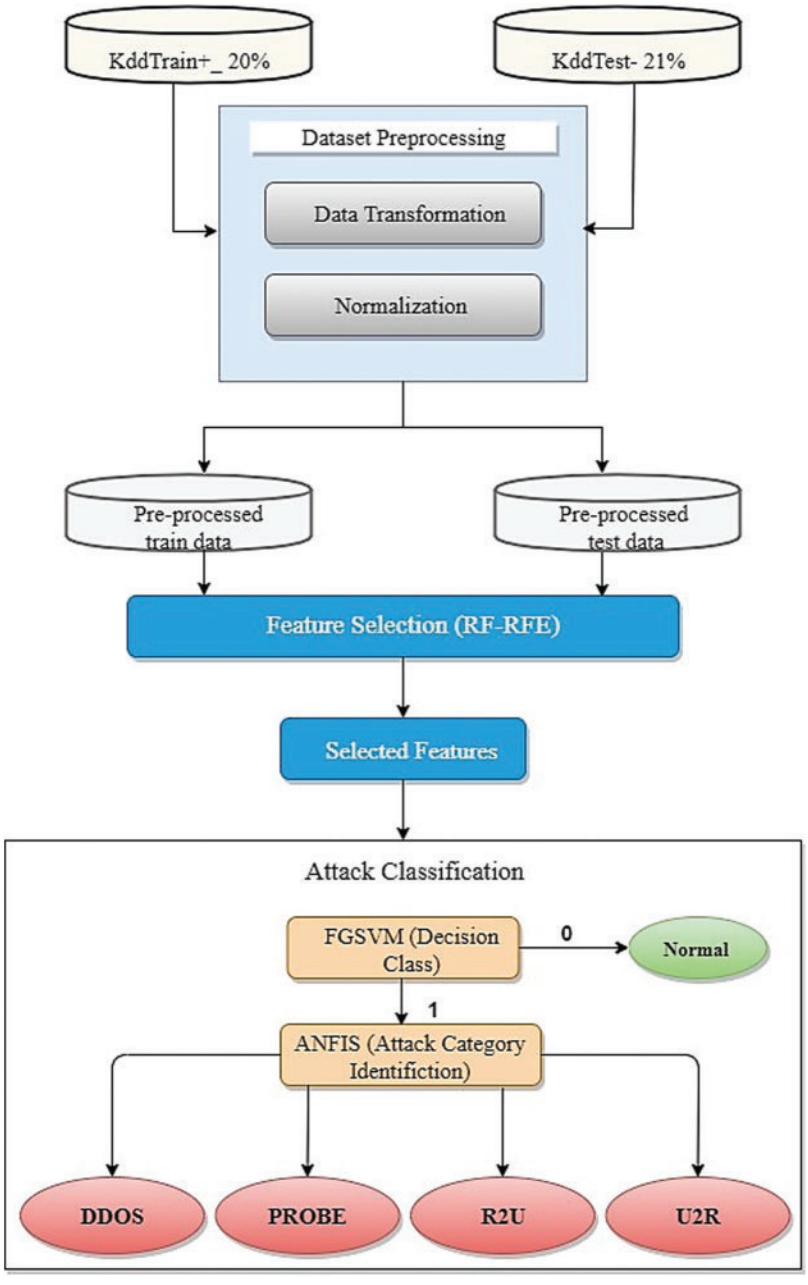

Figure 1: Proposed approach for intrusion detection

3.1 Dataset Selection

DARPA99 is the first data set created for the intrusion detection system at Lincoln laboratory in 1998. KDD99is the enhanced version of DARPA. Since the elimination of duplicate records in the KDD99 set of data has a substantial effect on the output of NSL-KDD [26] systems, the data set is considered a standard for the identification of Trespass in organizational structures. The usage of the NSL-KDD data set has the following benefits. Fig. 2 is a Visualization of some attributes of the NSL-KDD Data Set.

- Results are not biased because the train data set have no redundant records.
- Significant degradation rates owing to the lack of redundant information in the test sample.
- The chosen record is inversely proportional to each particular class category in the original KDD data collection.

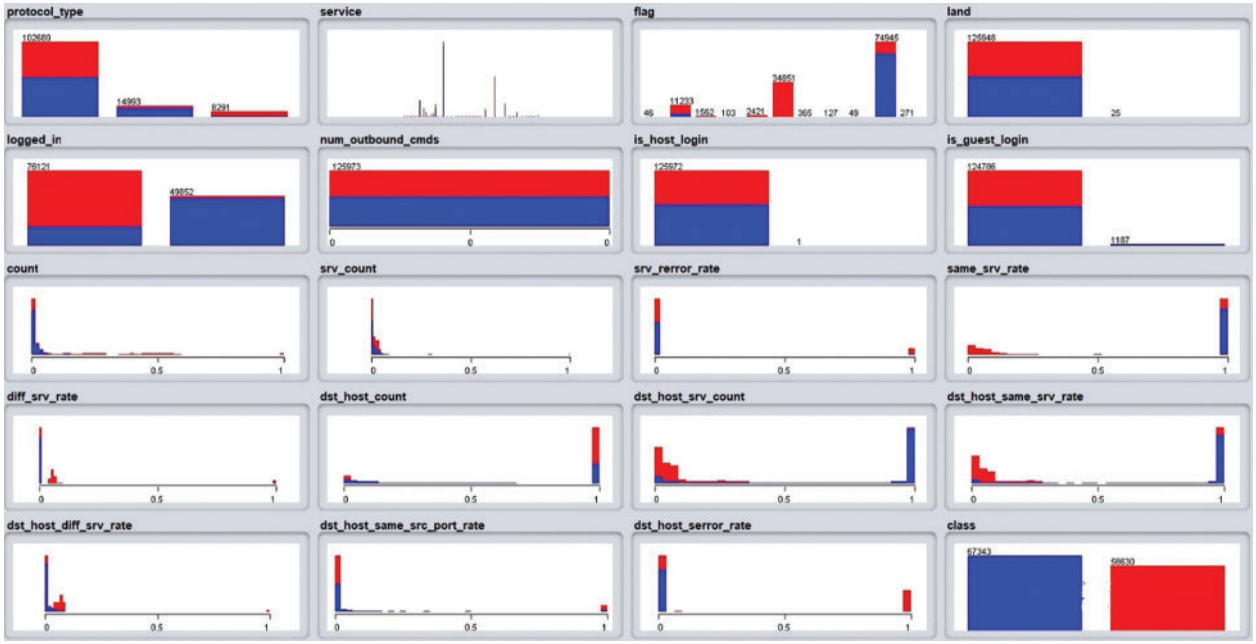

Figure 2: Visualization of some attributes in NSL-KDD data

3.2 Dataset Preprocessing

Data preprocessing is an important task to achieve accurate results through classifiers. Preprocessing of the original NSL-KDD data set is performed in MATLAB, which renders it an acceptable input to the classifier. Various data preprocessing techniques are available, but we use the following: Dataset Transformation and Dataset Normalization.

a) Data Transformation

In data transformation of nominal features to numeric value has been done. All the categorical values of attributes are transformed into numeric form. We assign random numbers to them. NSL-KDD Train 20% data set to have approximately 25192 Connection instances. Each connection instance has 42 features, including attack type. Value to the target class has also been

assigned in the same way. Zero is set for the normal class, and one will be for attack or any other deviation.

b) Data Normalization

Data set normalization is necessary when we have a large dataset that includes thousands of rows in training and testing files. The NSL-KDD dataset is large, so for accurate results of the classifier for intrusion detection. In this paper, we used the Min-Max method for the normalization of the dataset.

3.3 Feature Selection

Feature selection techniques are built to eliminate the number of input variables that are considered most appropriate for the model to predict the target variable. Random Forest (RF-RFE) is a machine-learning approach that could be perfect for integrating omics data commonly works well for high-dimensional problems and can classify strong predictors of a given outcome without making assumptions about the underlying model. Tree considers a different subset of randomly selected predictors, of which the best predictor is selected and split on at each node. In recursive features, elimination features are rank according to their importance score in a particular context. The significance of each feature is calculated at each iteration, and features with less importance are eliminated. Over a different subset of features, the importance of the same 46 feature can significantly vary while f evaluating highly correlated features. Protocol-type, logged-in, error-rate, srv-s error-rate, same-srv-rate are the features that are selected after applying the RF-RFE method.

3.4 Classification Process

SVM is a statistically supervised machine learning method that has been commonly used in the last few years. Classifies data inputs to various groups by creating a hyperplane. To conduct binary classification on data, SVM used a large dimensional space to figure out the hyperplane. The goal of optimizing the separation margin between the two groups is to find a hyperplane. SVM is used to identify input features into two distinct normal and attack classes. A line hyperplane separates a two-dimensional linearly separable input training data. The general function for the line is one Eq. (1):

$$y - ax + b \tag{1}$$

By replacing x with y1 and Y with y2 in 1 so we get Eq. (2):

$$ay1 - y2 + b = 0 \tag{2}$$

If we define Y= (y1, y2) and c = (a, −1) so 2 become Eq. (3):

$$c.y + b = 0 \tag{3}$$

The data points above on the hyperplane consider class [+1], and the points that are smaller from the hyperplane are considered class [−1], shown in Eq. (4):

$$h(yi) = \begin{cases} +1 \; if \; y.c + b \geq 0 \\ -1 \; if \; y.c + b < 0 \end{cases} \tag{4}$$

The function of the hypothesis is in Eq. (4) h (yi). When the hyperplane is constructed, it makes predictions using the Eq. above.

ANFIS is the second technique used in this process, based on the Takagi-Sugeno-fuzzy inference system. Due to the overlapping values among the pattern, ANFIS helps to check the dataset to eradicate vagueness. It will test the uncertainty between probe, U2R, DDOS, and R2U patterns, and their fuzziness will be eliminated. Input and output are two vectors that are used to train ANFIS. Premise functions for membership function may be identified using only a training dataset.

Let us assume that 'a' and 'b' are two input values for fuzzy and 'c' output. Two basic if-then rules for the first-order Sugeno model are followed by Eqs. (5)–(8):

Rule1: If a is A1 and b is B1 then, f1 = p1a + q1b + r1

Rule2: If a is A2 and b is B2 then, f2 = p2a = q2b = r2

$$f1 = p1a + q1b + r1 \tag{5}$$

$$f2 = p2a = q2b = r2 \tag{6}$$

$$W = w1f1 + w2f2/W1 + W2 \tag{7}$$

$$W^0 1f1 + W^0 2f2/W1 + W2 \tag{8}$$

Layer 1: Each nth node throughout the layer is just an adaptive node, and the node function in the layer is O1; n seems to be the output node in layer 1.

$$O1, n = \mu Xn(a), n = 1, 2, \tag{9}$$

$$O1, n = \mu Yn - 2(b), n = 3, 4 \tag{10}$$

In Eqs. (9) and (10), a, b are the inputs to node ith node at layer 1. Ai and Bi are the linguistic variables.

$$\mu X(a) = 1/1 + |a - cn/an|2b \tag{11}$$

ai and ci are the parameters in 15 is set with the change in these values bell shape function also changes. Layer 2: Each node within that layer is labeled as ANFIS, and the fixed node output is created by taking the product of all the coming signals as shown in Eq. (12):

$$O2, n = wn = \mu Xn(a)\mu Yn(b), n = 1, 2 \tag{12}$$

The output of each node represents a rule firing strength.

Layer 3: The ratio of nth rule n firing power to the total of all rules firing power is measured at the nth node, and the output at layer 3 is often referred to as normalized firing power in Eq. (13).

$$O3, n = w^0 = wn/w1 + w2, n^0 = 1, 2 \tag{13}$$

Layer 4: At layer 4, each node n is an adaptive node with the function of the b node.

$$O4, n = wifnapt = wapt(pna + qnb + rn) \tag{14}$$

In pn, qn, rn in Eq. (14) are the corresponding parameters that are placed.

Layer 5: The cumulative output shown in Eq. (15) is determined by summing up all incoming signals within layer 5.

$$O5, n = \sum Iwnfn = \sum nwn2 / \sum nwn \tag{15}$$

4 Experimental Analysis and Results

First data is categorized into two distinct normal and attack classes by using the FGSVM classifier. Compared with other traditional classification techniques, FGSVM gives the best accuracy in the classification of data with fine-tuning. Thus, nonlinear separation can be accomplished in the feature space as the FGSVM discovers a hyperplane in the portion space.

4.1 Performance Measures

FGSVM detects the attack and normal classes in the NSL-KDD dataset with the help of classification techniques so, accuracy, recall Precision, and F-score are performance metrics. Tab. 2 showed some accuracy measure and their description. Tab. 3 shows the terms to visualize the performance of the confusion matrix of Fig. 3. A prediction summary of results on the classification problem is given in the confusion matrix. The numbers of true and false predictions done by the classifier are summarized with count values. Fig. 4 shows the ROC curve. Recall or the true positive rate is how many true positives attack classes get predicted from all the dataset's network records. It is sometimes also called sensitivity. Recall The false-positive rate is a plot against the true positive rate in the receiver operating curve. A model's accuracy based on recall and Precision can be measure by finding an f-score. The usefulness of the test is compared by using the ROC curve.

Table 2: Performance analysis of the proposed work

Classifier	Performance measure	Formula	Value %
FGSVM	Accuracy	TP + TN/(TP + TN) + (FP + FN)	99.3
	Specificity	(TN)/(TN + FP)	0.998
	Precision	TP/(TP + FP)	0.999
	Recall	TP/TP + FN	0.992
	F-measure	2 × (recall* Precision)/(recall Precision)	0.995

Table 3: Terms to visualize performance of confusion matrix

True positive (TP)	The network found with intrusion detection
False positive (FP)	Network incorrectly classifies as anomaly-based
True negative (TN)	Network successfully describe no intrusion
False negative (FN)	An intruder is in a network but couldn't be identified by classification

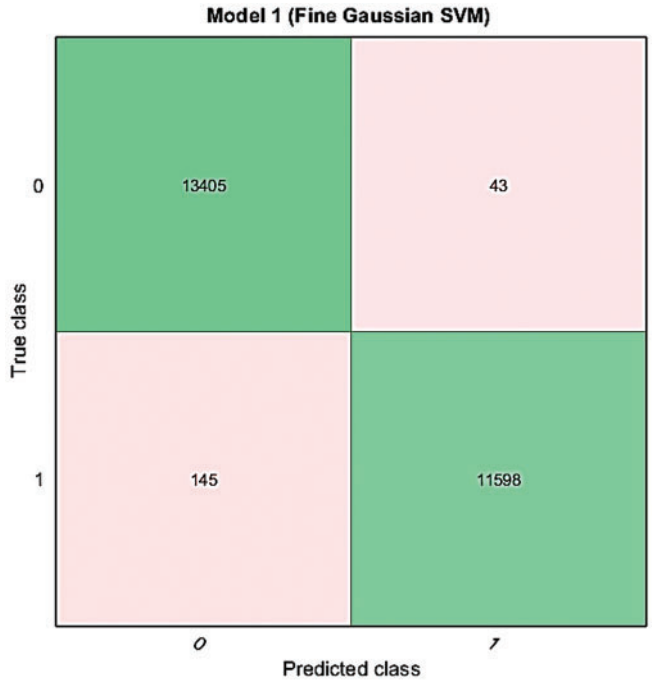

Figure 3: Confusion matrix

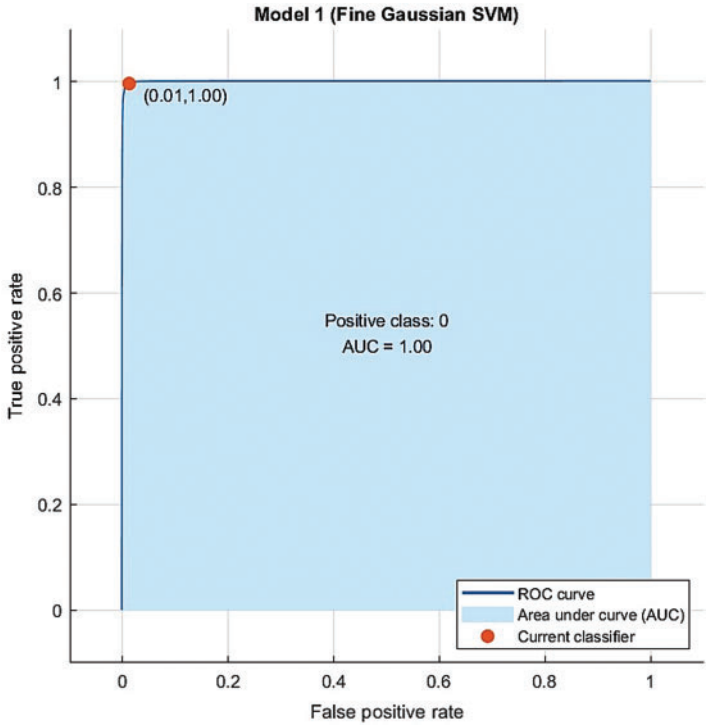

Figure 4: ROC curve

Accuracy given in Tab. 3 is how particular classification techniques perform well at a classification of data. FGSVM gave 99.3% accuracy in the classification of data into the attack and

normal class. A recall is a total number of correctly classified positive data divided by the entire positive data. The higher the value of recall higher the accuracy rate of the predicted class. The Recall value which is attained in the FGSVM model is 0.992%. The Precision shows that the positive values in data after classification are labeled as positives. The Precision of FGSVM is 0.99% which shows 99% of data is accurately classified as attack class. Precision and sensitivity weighted average is measure in F-measure. Classified abnormal Patterns are separated from the dataset, and the feature selection method is applied before input data to fuzzy. RF-RFE method is used to select essential predictors from the dataset. Fig. 5 represents the predictors according to their importance by constructing many decision trees. Features are ranked according to their importance scores attained after running RF once. After running it recursively, feature importance is measured, and the less relevant one is removed at each iteration. The most relevant features that are selected are protocol-type, logged-in, error-rate, SRV-error-rate, same-SRV-rate.

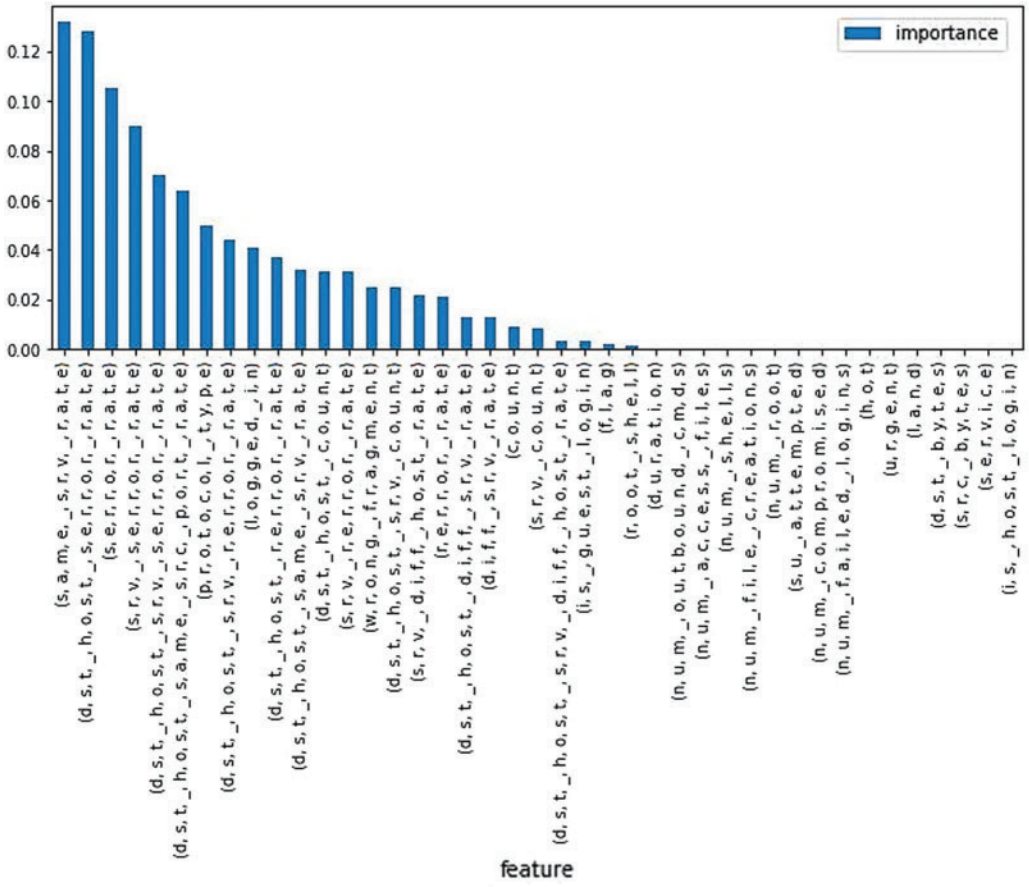

Figure 5: Features ranking based on their importance

4.2 Implementation of ANFIS

Data is classified into normal and attacked classes using fine Gaussian SVM. To deal with computational uncertainty, ANFIS is used, which includes the degree of truthiness instead of conventional Boolean Logic. Fig. 6 shows the ANFIS structure to eliminate the inherent fuzziness

of the patterns classified through SVM. ANFIS is used for the exact identification of probe, U2R, R2U, and DDOS. We have used 20% of the dataset as training data, test 21% for testing, and kept 15% for checking. Finally, the exact identification results can be seen through the surface viewer that illustrates the significance of two input variables and their effect on output in Fig. 7.

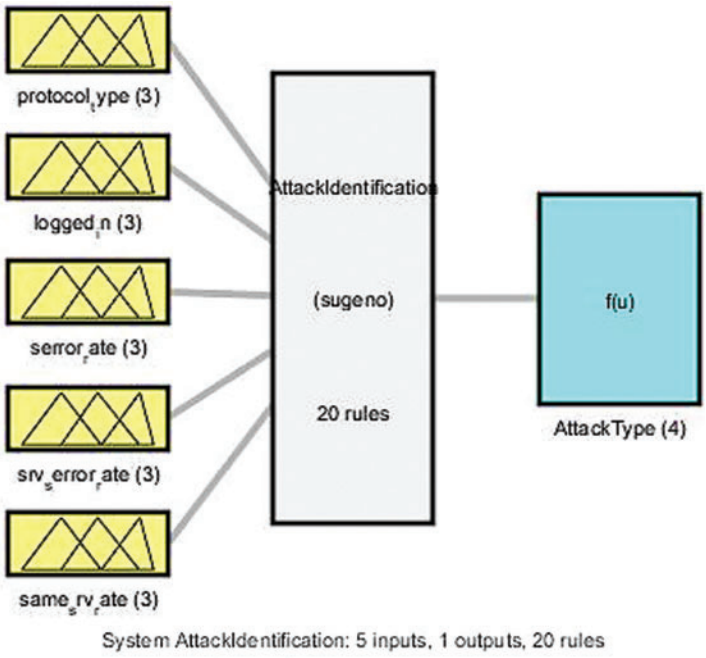

Figure 6: Fuzzy inference system

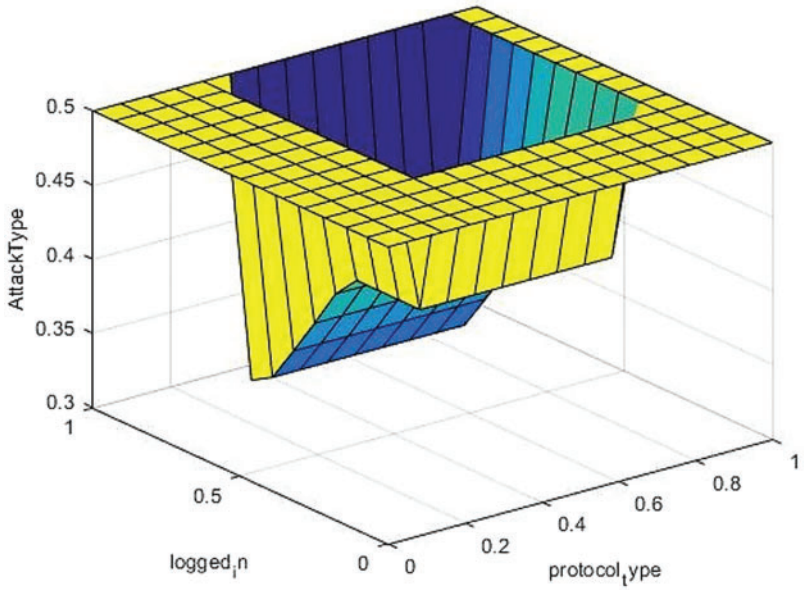

Figure 7: Surface viewer

4.3 Fuzzy Rules

The rule base seems to be the characterizing aspect of the fluffy reasoning approach, and the Precision and quality of the methodology depend on the fluffy criteria. Some of the rules that are specified for detecting attacks are as followed in Tab. 4:

(1) If (protocol-type is high) and (logged-in is medium) and (error-rate is high) and (SRV-error-rate is medium) and (same-SRV-rate is medium) then (AttackType is DDOS) (1)
(2) If(protocol-type is high) and (logged-in is medium) and (serror-rate is high) and (srv-serror-rate is high) and (same-srv-rate is low) then (AttackType is DDOS) (1)
(3) If (protocol-type is medium) and (logged-in is high) and (serror-rate is high) and (srv-serror-rate is medium) and (same-srv-rate is medium) then (AttackType is U2R) (1)
(4) If (protocol-type is medium) and (logged-in is medium) and (serror-rate is low) and (srv-serror-rate is medium) and (same-srv-rate is high) then (AttackType is U2R) (1)
(5) If (protocol-type is medium) and (logged-in is low) and (serror-rate is high) and (srv-serror-rate is medium) and (same-srv-rate is medium) then (AttackType is R2L) (1)
(6) If (protocol-type is high) and (logged-in is medium) and (serror-rate is medium) and (srv-serror-rate is high) and (same-srv-rate is high) then (AttackType is R2L) (1)
(7) If (protocol-type is high) and (logged-in is medium) and (serror-rate is medium) and (srv-serror-rate is high) and (same-srv-rate is high) then (AttackType is R2L) (1)
(8) If (protocol-type is medium) and (logged-in is medium) and (serror-rate is low) and (srv-serror-rate is medium) and (same-srv-rate is low) then (AttackType is Probe) (1)
(9) If (protocol-type is low) and (logged-in is high) and (serror-rate is medium) and (srv-serror-rate is medium) and (same-srv-rate is high) then (AttackType is Probe) (1) These are one of few rules out of 46 that are defined in fuzzy to detect either the coming traffic is normal or malicious.

Table 4: Ranges of MF for input variables

μ (Protocol-type)	$= \{0 < x < 0.597 = \text{low}, 0.299 < x < 0.883 = \text{medium}, [0.602 < x < 0.993 = \text{high}\}$
μ (logged-in)	$= \{0 < x < 0.597 = \text{low}, 0.299 < x < 0.883 = \text{medium}, [0.602 < x < 0.993 = \text{high}\}$
μ (same-srv-rate)	$= \{0 < x < 0.597 = \text{low}, 0.299 < x < 0.883 = \text{medium}, [0.602 < x < 0.993 = \text{high}\}$
μ (serror-rate)	$= \{0 < x < 0.597 = \text{low}, 0.299 < x < 0.883 = \text{medium}, [0.602 < x < 0.993 = \text{high}\}$
μ (srv-count-rate)	$= \{0 < x < 0.597 = \text{low}, 0.299 < x < 0.883 = \text{medium}, [0.602 < x < 0.993 = \text{high}\}$

The proposed technique's findings were addressed in this segment utilizing the alternative (hybrid) solution to identifying attack identification in an organization that has been carried out. One of the main obstacles to the current solution is to identify data and then detect attacks accurately. Preprocessing on dataset increase SVM accuracy rate, and the SVM accuracy rate for classification of the NSL-KDD dataset is 99.3%.

5 Conclusion

In this research, a detection system is projected on the NSLKDD dataset by applying data transformation and maximization and minimization methods. FGSVM is used to classify the NSLKDD dataset into two classes normal class and attack class. Substantial results obtained

from FGSVM have shown 99.03% to identify DDOS, probe, U2R, and R2L. FGSVM identified abnormal patterns are stimulated through ANFIS. According to their importance scores and prediction role, five features are selected for training, testing, and validation procedures of ANFIS. Error tolerance and epochs are set from zero and 100. During training, the MSE is 0.08523, and on testing and validation, the MSE is 0.08496, which reflects reasonable accuracy rates for the precise identification of DDOS, Probe, R2U, and U2R. To find out the intrusion and to prevent the network from it, ANFIS quickly build previously extracted connections and record-based measures as it is a rule-based method. FGSV'M and ANFIS can be robust potential solutions that significantly boost their efficiency as more machine learning. In the future, deep learning-based classifiers are available to classify data and make systems more accurate and cost-effective. Advanced machine learning and AI-based systems are improving in intrusion detection with a higher accuracy rate.

Acknowledgement: The authors would like to thank the Deanship of Scientific Research at Prince Sattam bin Abdul-Aziz University, Saudi Arabia.

Funding Statement: The authors would like to thank the Deanship of Scientific Research at Prince Sattam bin Abdul-Aziz University, Saudi Arabia.

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

References

[1] R. Heady, G. Luger, A. Maccabe and M. Servilla, "The architecture of a network level intrusion detection system," Tech. Rep., Los Alamos National Lab, New Mexico University, Albuquerque, NM (United States), 1990.

[2] S. Bhattacharya, P. K. R. Maddikunta, R. Kaluri, S. Singh, T. R. Gadekallu *et al.* "A novel PCA-firefly based XGBoost classification model for intrusion detection in networks using GPU," *Electronics,* vol. 2, no. 19, pp. 219, 2020.

[3] RM, S. Priya, P. K. R. Maddikunta, M. Parimala and S. Koppu, T. R. Gadekallu *et al*. "An effective feature engineering for DNN using hybrid PCA-GWO for intrusion detection in IoMT architecture," *Computer Communications,* vol. 160, pp. 139–149, 2020.

[4] H. Debar, "An introduction to intrusion-detection systems," *Proc. of Connect,* vol. 2000, 2000.

[5] S. Namasudra, "Fast and secure data accessing by using dna computing for the cloud environment," *IEEE Transactions on Services Computing,* 2020.

[6] S. Namasudra, R. Chakraborty, A. Majumder and N. R. Moparthi, "Securing multimedia by using DNA-based encryption in the cloud computing environment," *ACM Transactions on Multimedia Computing, Communications and Applications (TOMM),* vol. 16, no. 3s, pp. 1–19, 2020.

[7] S. Kumari and S. Namasudra, "System reliability evaluation using budget constrained real d-mc search," *Computer Communications,* vol. 171, pp. 10–15, 2021.

[8] S. Kumari, R. J. Yadav, S. Namasudra and C.-H. Hsu, "Intelligent deception techniques against adversarial attack on the industrial system," *International Journal of Intelligent Systems,* vol. 36, no. 5, pp. 2412–2437, 2021.

[9] P. Pavithran, S. Mathew, S. Namasudra and P. Lorenz, "A novel cryptosystem based on dna cryptography and randomly generated mealy machine," *Computers & Security,* vol. 104, pp. 102160, 2021.

[10] F. Sabahi, A. Movaghar "Intrusion detection: A survey," in *The Third Int. Conf. on Systems and Networks Communications*, 2008.

[11] S. Li, G. Wang and J. Yang, "Survey on cloud model based similarity measure of uncertain concepts," *CAAI Transactions on Intelligence Technology,* vol. 4, no. 4, pp. 223–230, 2019.

[12] R. M. Alguliyev, R. M. Aliguliyev and L. V. Sukhostat, "Efficient algorithm for big data clustering on single machine," *CAAI Transactions on Intelligence Technology,* vol. 5, no. 1, pp. 9–14, 2020.

[13] M. Alazab and M. Aiash, "Machine learning based botnet identification traffic," in *2016 IEEE Trustcom/BigDataSE/ISPA*, IEEE, pp. 1788–1794, 2016.

[14] M. Alazab, R. Layton, R. Broadhurst and B. Bouhours, "Malicious spam emails developments and authorship attribution," in *2013 Fourth Cybercrime and Trustworthy Computing Workshop, IEEE*, pp. 58–68, 2013.

[15] U. A. Sandhu, S. Haider, S. Naseer and O. U. Ateeb, "A survey of intrusion detection & prevention techniques," in *2011 Int. Conf. on Information Communication and Management, IPCSIT*, vol. 16, pp. 66–71, 2011.

[16] P. Miksovsky, K. Matousek and Z. Kouba, "Data preprocessing support for data mining," in *IEEE Int. Conf. on Systems, Man and Cybernetics*, IEEE, vol. 5, pp. 4, 2002.

[17] J. S. Malik, P. Goyal and A. K. Sharma, "A comprehensive approach towards data preprocessing techniques & association rules," in *Proc. of the 4th National Conf.*, vol. 132, 2010.

[18] F. S. Girish Chandrashekar, "A survey on feature selection methods," *Computers and Electrical Engineering,* vol. 24, pp. 16–28, 2014.

[19] S. N. Samina Khalid and Tehmina Khalil, "A survey of feature selection and feature extraction techniques in machine learning," in *2014 Science and Information Conf.*, London, UK, vol. 24, pp. 1–13, 2017.

[20] B. Ingre, A. Yadav and A. K. Soni, "Decision tree based intrusion detection system for NSL-KDD dataset," in *Int. Conf. on Information and Communication Technology for Intelligent Systems*, Springer, pp. 207–218, 2017.

[21] S. Aljawarneh, M. Aldwairi and M. B. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," *Journal of Computational Science,* vol. 25, pp. 152–160, 2018.

[22] 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.

[23] U. Ahmad, H. Asim, M. T. Hassan and S. Naseer, "Analysis of classification techniques for intrusion detection," in *2019 Int. Conf. on Innovative Computing*, New Delhi, India, IEEE, pp. 1–6, 2019.

[24] S. A. Aziz, E. Sanaa and A. E. Hassanien, "Comparison of classification techniques applied for network intrusion detection and classification," *Journal of Applied Logic,* vol. 24, pp. 109–118, 2017.

[25] A. Hajimirzaei and N. J. Navimipour, "Intrusion detection for cloud computing using neural networks and artificial bee colony optimization algorithm," *ICT Express,* vol. 5, no. 1, pp. 56–59, 2019.

[26] M. R. Parsaei, S. M. Rostami and R. Javidan, "A hybrid data mining approach for intrusion detection on imbalanced NSL-KDD dataset," *International Journal of Advanced Computer Science and Applications,* vol. 7, no. 6, pp. 20–25, 2016.

[27] Y. S. Sydney and M. Kasongo, "A deep learning method with wrapper based feature extraction for wireless intrusion detection system," *Computers & Security. Elsevier,* vol. 92, pp. 15, 2020.

[28] K. E. S. Hadeel Alazzam and Ahmad Sharieh, "A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer," *Expert Systems with Applications. Elsevier,* vol. 148, pp. 113249, 2020.

[29] A. M. Yogita Hande, "A survey on intrusion detection system for software defined networks (sdn)," *Research Anthology on Artificial Intelligence Applications in Security. IGI Global,* vol. 16, no. 1, pp. 20, 2021.

[30] A. R. Javed, M. O. Beg, M. Asim, T. Baker and Al-Bayatti, "Alphalogger: Detecting motion-based side-channel attack using smartphone keystrokes," *Journal of Ambient Intelligence and Humanized Computing,* pp. 1–14, 2020.

[31] A. R. Javed, Z. Jalil, A. Moqurrab, S. Abbas and X. Liu, "Ensemble adaboost classifier for accurate and fast detection of botnet attacks in connected vehicles," *Transactions on Emerging Telecommunications Technologies,* pp. e4088, 2020.

[32] S. I. Imtiaz, S. Rehman, A. R. Javed, Z. Jalil, X. Liu and W. S. Alnumay, "DeepAMD: Detection and identification of android malware using high-efficient deep artificial neural network," *Future Generation Computer Systems,* vol. 115, pp. 844–856, 2021.

[33] A. Basit, M. Zafar, X. Liu, A. R. Javed, Z. Jalil *et al.*, "A comprehensive survey of AI-enabled phishing attacks detection techniques," *Telecommunication Systems,* vol. 76, pp. 1–16, 2020.

[34] A. Rehman, S. U. Rehman, M. Khan, M. Alazab and T. Reddy, "CANintelliIDS: Detecting in-vehicle intrusion attacks on a controller area network using CNN and attention-based GRU," *IEEE Transactions on Network Science and Engineering,* pp. 1–1, 2021.

[35] S. Sharma, S. Ahmed, M. Naseem, W. S. Alnumay, S. Singh *et al.*, "A survey on applications of artificial intelligence for Pre-parametric project cost and soil shear-strength estimation in construction and geotechnical engineering," *Sensors,* vol. 21, no. 2, pp. 463, 2021.