Network Anomaly Detection using Predictive Analysis in Machine Learning

MSc Research Project
Data Analytics

Ritu Verma
Student ID: X18181040

School of Computing
National College of Ireland

Supervisor: Christian Horn

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Ritu Verma
Student ID:	X18181040
Programme:	Data Analytics
Year:	2019-20
Module:	MSc Research Project
Supervisor:	Christian Horn
Submission Due Date:	17/08/2020
Project Title:	Network Anomaly Detection using Predictive Analysis in Machine Learning
Word Count:	XXX
Page Count:	24

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	27th September 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Network Anomaly Detection using Predictive Analysis in Machine Learning

Ritu Verma
X18181040

Abstract

With immense growth and rapid rise in detection of intrusion, undoubtedly it plays a key role in the security of existing systems. The present approaches available in the systems for detection of intrusion are somehow adequate and less effective to an extent. Many conventional approaches to accentuate (IDS) Intrusion Detection system claims an artificial neural network to be better in comparison to traditional methods. However, the strategies based on ANN require enhancement, exceptionally for less frequent attacks. In this research, a novel approach based on ANN (artificial neural network) and genetic algorithm for feature selection with optimal number of feature value are proposed. This new approach is proposed to achieve better accuracy and resolve the problem by aiming to gain more stability with a less false positive rate for the detection system. Results achieved through experiments on NSL KDD dataset demonstrate that the proposed approach in this paper, outer-performs the existing esteemed methods concerning false -positive rate and accuracy.

Keywords : ANN (Artificial neural network), Back propagation algorithm, genetic algorithm, random forest, recursive feature elimination ,anomalies.

1 Introduction

With the immense and rapid growth of technologies especially the internet in the industries leads to a rise in the development across the industries but at the same point of time also chances of security vulnerabilities are increasing with the movement and change of platform within applications and the internet. With the rise in cyber-attacks over recent years particularly against industries and companies with internet service has become a prominent security issue as its vital for any business to protect and secure their data and assets. (Chaudhary et al.; 2019) Web applications for instance online shopping etc.. with internet advent in a way have provided a platform to network security to build a strong pillar. Analysation of data in process of the network to prevent attacks by identifying intrusion plays a crucial role. (Sani et al.; 2009) There are times with unexpected fall occurs in applications that lead to major impact on operations of business throughout downstream and online and the average downtime cost approximately is \$100, 000/hour. (Kromkowski et al.; 2019) So, detection of attacks or recovering threats cannot be sufficient in spite there must be rapid reactionary abilities in terms of detection of attacks.

In principle to the aforementioned statement, Intrusion detection systems based on anomalies can recognize data packets in traffic of network and analyze the ones which do not to the normally generated profile. ANN(Artificial neural network) in the neural network processes the incoming data and further send to an expert system. There are two regular technologies in IDS, signature search and rule-based, where on one side signature-based is easy to be configured but are not best at recognizing unknown attacks due to time is taken in a continuous update of the database whereas anomaly-based systems are more flexible and recognize attacks with life span. An artificial neural network is also used as an alternative in context to statistical learning, the system of detection for anomalies. As the rule-based approach is based on a pre-defined set of rules and this approach is mostly used by traditional solutions so this system requires upgrades as the current one does not recognize scenarios of threats. So, related to this purpose for detecting new and changing threats artificial neural networks are quite successful and they play a prominent role in respect to the security of services.

In this research, in the first approach, as the NSL-KDD dataset is highly skewed with unequal distribution of samples among all five types of connections due to which there has been much-declined accuracy in the detection of R2L and U2R attacks so this problem is reposed as a Binary classification problem, rather than a 5-class detection problem. Merging all four attacks as one namely ‘anomaly’ helped to achieve a fairly balanced distribution on which detectors can be trained efficiently. In the later approach, which is the core approach of this research paper, to overcome the problems of network anomaly detection, the problem is formulated as a Binary classifier problem – classifying as ‘anomaly ‘ or ‘normal’ depending upon 41 attributes as input. Firstly, the feature set is expanded and then ANN is trained. Further to it, through a genetic algorithm, a heuristic search was performed to get the optimal number of features to be retained. 70 features were found to be an optimal number. In total, 3 feature selection algorithms - Genetic algorithm, RFE, and SelectKbest, were applied to reduced feature vectors to a size of 70, and 3 more ANNs were trained. Now, the performances of all these 4 ANNs were compared upon test set using F1-score, the area under ROC, etc. We found ANN trained upon Genetic Algo-reduced features to be best. We prepared that as the final classifier (designated as ’BEST detector’ here) and present its final test set performance detection system for anomalies is developed depending upon methods of ANN for network anomaly detection. Experiments on the dataset are conducted for evaluation of the performance of the model. The model presented in the paper outer performs with existing art-of-models in fields of false-positive rate and accuracy.

2 Related Work

Over the period of time, detection systems on network anomaly have been established, however, with the rise in technologies, the range of attack types are also increasing. So, to overcome, various methods, systems, and tools have been configured to obtain the best detector which can give an optimal number of features.

2.1 Artificial Neural Network-based model

(Jones et al.; 2018) has proposed a self-regulating based solution as while detection of network-based attacks, it gives reactive responses. In the solution, the author has provided a module to detect attacks, and based on unsupervised ANN reacts to the network which is monitoring including attacks. The author has identified two scenarios, without any specific false positives. However, there could be more ways to predict attacks.

Cahyo et al. (2016) implemented a fortuitous study of ANN(artificial neural network) and SVM (support vector machine) with different types of attacks. The results achieved after experiments on the study proved that ANN performs better over SVM majorly on the detection of four types of attacks – remote to local (R2L), the user to Root(U2R), probe, and denial of service(DoS). Noticeable observation through this paper is ANN's performance is outstanding for probe attack and R2L. Almost double accuracy achieved in the detection of R2L is remarkable. So, it highlights that ANN can achieve better accuracy in comparison to other data mining algorithms for instance k-means clustering, Naïve Byes, SVM, etc.. However, one drawback that was the only one observed was scarcity in the capability of the system to show data representations. As anomaly could be a better achiever by applying feature extraction on normalization in combination with various machine learning algorithms because feature extraction would help for better enhancement of the process for better learning of data representations.

(Subba et al.; 2016) feedforward and back-propagation algorithm is used to build ANN, the model consists of input, hidden, and output layer. The number of nodes in input and hidden layers is synched as per the quantity of vector of input feature and number of features in it.

However, results on the NSL-KDD dataset showcase that it is outstanding for Naïve Byes but its computational head is less in comparison to the ANN model which used only individual hidden layers. Various parameters of the intrusion detection system have to be tuned in the future work of this paper.

(Sahu and Mukherjee; 2020) have used two machine learning models which are classification models and they have compared the performance of both the models and presented them in their study. Artificial neural networks and logistic regression are the two models used in this paper. For ANN, 99.4% accuracy is achieved, and by using logistic regression 99.99% accuracy is obtained. Hence, this work can be used in IoT solutions and smart devices to prevent attacks.

2.1.1 Genetic Algorithm with ANN

(Punitha et al.; 2019) As, in the intrusion detection system due to dimensionality time complexity increases and reduction in resource utilization occurs which leads to a high number of comparisons. So, to overcome this problem genetic algorithm as a feature ranking technique is used. Ranks that are achieved from individual correlation and data gain have been combined to get feature reduction. So, a novel approach is used to detect both useless and helpful options. The feature is taken as input which produces the best output set through a genetic algorithm. Overall, the system does a few limited quantities of comparisons which results in a high-performance rate with respect to classification accuracy.

In all of the techniques, it is significant to observe that the PCA approach was one of the most successful ones which lead to accuracy enhancement. The level of accuracy was enhanced from 80.31% to 88.74%. However, with lower FAR also better accuracy can be achieved along with other features and methods of machine learning.

2.2 Machine learning supervised models

Naseer et al. (2018) have built a model to detect intrusion, it was trained along with the proposed architecture of the deep neural network. Decision trees, random forest, and support vector machines as a classification technique have been used. Classification metrics along with classification techniques have been used. Further to this, the effect of encoding schemes on the dataset of NSLKDD is analyzed by using a convectional classifier. Thus, the model proves to achieve higher accuracy of 85% on the test dataset. So, this proves that deep learning being feasible and promising technology is also secured for information applications. However, the only drawback of the study was they did not involve any troll for the extraction of feature hence raising the signal for a need to have a tool which could be capable enough to understand efficient data to resolve issues of anomaly detection.

Bitaab and Hashemi (2017) has built a hybrid system using a decision tree for intrusion detection and using it as a component of misuse detection for detection of anomalies is highlighted. Training data consists of known and normal attacks where the decision tree identifies known attacks and it stops labeling as normal for upcoming new instances, it labels them as normal if it is an unknown attack so to overcome the issue GMM is used to for leaf distribution and achieved false-positive 9.37% and 96.72% as detection rate and accuracy as 94.10%.

2.3 Deep Neural Network-Based Model

For huge datasets, deep neural network models are applied so that outstanding results can be achieved. Most importantly, if the dataset is used for intrusion detection consists of the highest number of instances then the best approach for attack classification is deep learning.

Naseer et al. (2018) targets to find the necessity of techniques of deep learning for detection of anomalies To conquer the issue of overfitting author does the prediction of thin subsets and trains the ensemble network and applies the techniques considering every member as a subclass of the main neural network and uses all 41 features in such a form that is absorbed by DNN's. The author also uses cross-validation techniques and trained deep models on NSLKDD, however, due to the cross-validation technique overfitting occurs which the author solves by the aforementioned approach. Finally, the authors suggest that deep learning can be used as a tool for feature extraction in future work for anomaly detection.

2.4 SVM

(Zhang et al.; 2019) have proposed anomaly-based SVM to tackle the issues with the existing techniques of SVM's in the present approaches of SVM with training feature are not able to detect short-duration attacks and intrusions so efficiently in the traffic. So, in their research, detection scheme of SVM based on anomaly is built by optimizing and extracting training features. Calculation of SVM is done by two features data plane traffic and packet count control. Cross-correlation, KL divergence, and linear function helps to enhance the accuracy of detection efficiently and to detect intrusions which are of short duration.

(Primartha and Tama; 2017) have used a random classifier for accuracy and performance improvement of anomaly detection. For evaluation purposes on the NSL KDD dataset, 10 classifiers were generated in accordance with the ensemble and the number of trees in it. Accuracy (91.8%) and FAR (6.35%) were achieved as RF -800 turned to be the most statistically significant factor in comparison with other classifiers.

2.5 K-fold cross validation

(Behera et al.; 2018) applied k-fold cross-validation and then the dataset is split into subsets and k times this method is repeated to overcome the problems of missing vital patterns which occurs as only the performance on training data could be monitored. K-fold adopts a split method as for testing it keeps the proportion of data. In every trial, k-1 subsets and one subset for testing are arranged for testing so that maximum data can be used in validation and data fitting as variance faces reduction. K= 10 value is set to obtain maximum efficiency, particularly in this study.

2.6 Naïve Bayes

(Chitrakar and Huang; 2012) has proposed a combination of k-means clustering and naïve byes algorithm. For minute data points, k-means are used and rapid execution as it is strong in this research, however, intrusions are not so accurately differentiated by k-means so naive byes is used to overcome the problem as naïve byes are used to analyze the relationship between dependent and independent variable through conditional probability. Hence, two divisions of the experiment were done and naïve byes was run allowed to run before k-means.

2.7 Back propagation algorithm

(Shah and Trivedi; 2015) has performed testing comparisons and basic N-fold validation on a reduced dataset which consists of all full features of a dataset. Through basic comparison, it can be observed that a reduced dataset performs outstandingly on time, size as well as complexity parameters. Further experiments on n-fold validation prove that when a reduced dataset is being used by classifiers then those classifiers turn to have improved generalization capacity. Additionally, during testing comparison, it was found that datasets are equally compatible.

3 Methodology

3.1 Dataset

(Meena and Choudhary; 2017) The dataset used is the NSL-KDD dataset.NSL-KDD is a smaller, cleaned, and polished version of the foremost KDD dataset. Testing and training subsets consist of 22544 and 125973 entries respectively. The entries illustrate the instances of the attribute of connection. The total quantity of attributes is 42. The categorization of connection instance is done as 'connection type ' attribute, it takes 32 and 23 types in testing and training samples respectively.

3.2 Data analysis

Step 1: Dataset is read into the Pandas data-frame and checked if any entry in the existing data frame is None.

Step 2: Step 2. The values acquired by individual 42 features are shown through the histogram plot to gain a better idea of the inadequacy of attributes throughout the samples. If the observed histogram is peaked or one-sided locally, then it depicts less significant attribute due to its narrow range For instance, attribute – the duration' is over 0.0 92% throughout the training samples.

Step 3: Step 3. Step 2 is repeated for all the 42 attributes for the elimination of the least significant features. The count of attributes tagged as *not_required* were 18 and the remaining 24 attributes were tagged as *required.*

Step 4: Two child data-frames containing *not_required* and *required* attributes were respectively subjected to computation of cross-correlation. *required* attributes show higher degree of cross-correlation amongst themselves than the *not_required* attributes. Further, the first-order statistics –count, mean, standard deviation, minima, maxima, 0.25, 0.50 & 0.75 percentile values are derived.

Note: 4 attributes namely, *protocol, service, flag, and connection_type* have non-numerical values, hence they were not taken in to consideration for cross-correlation and first order statistics analysis.

Step 5: Range of the 20 numeric-valued attributes are analyzed with respect to each of the non-numeric attributes- *protocol, flag, and connection_type* using box-plots.

The cross-correlation amongst the required attributes revealed a high degree of correlation while the *not_required* attributes had a very low degree of correlation amongst themselves. The box plots of *required* attributes with respect to the 3 selected attributes revealed a high degree of skewness and a very haphazard dispersion of the values. This suggested the inability of the first-order statistics to clearly model the distribution for developing any successful anomaly detector. Hence, we moved to the machine learning models which yield inference systems by exploring the higher-order statistics in a large feature-vector space.

Analyzing the individual histograms of all 38 numeric-valued attributes in the given dataset, it was evident that only 20-numeric-valued attributes span a broad range of values and, remaining 18 attributes which most frequently attained either a single value or just a few values, can be tagged as *not_required.*

The cross-correlation amongst the 20 *required* attributes reveals a high degree of correlation between the attributes as shown in the figure below. However, the 18 *not_required* attributes have a very low degree of correlation amongst themselves as shown in the subsequent figure.

3.3 Data pre-processing

Pre-processing aimed dominantly to deal with the 4 non-numeric valued attributes. Table 1 depicts the number of non-numeric values attained by these 4 attributes.

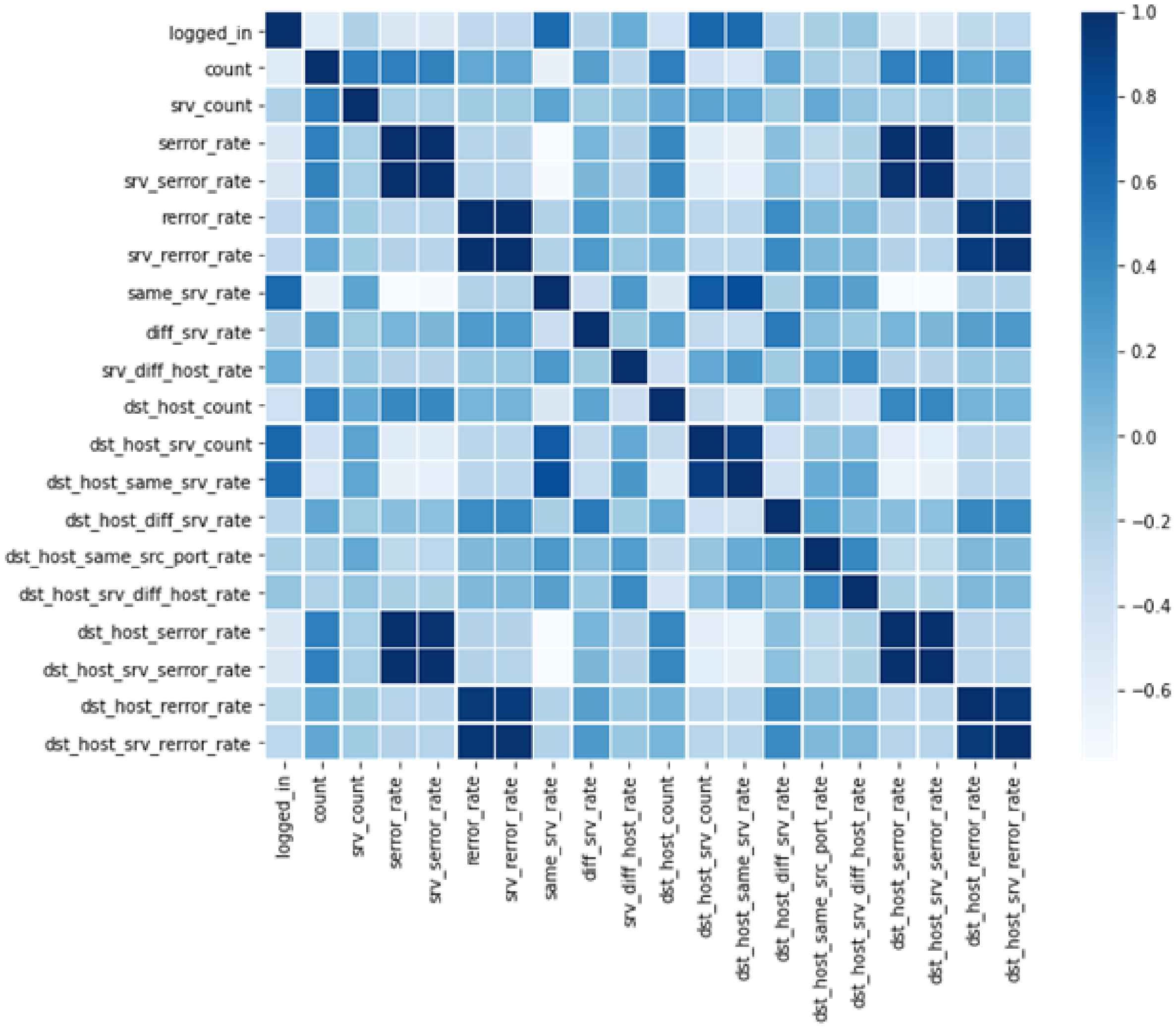


Figure 1: Cross-correlation of the 20 *required* attributes

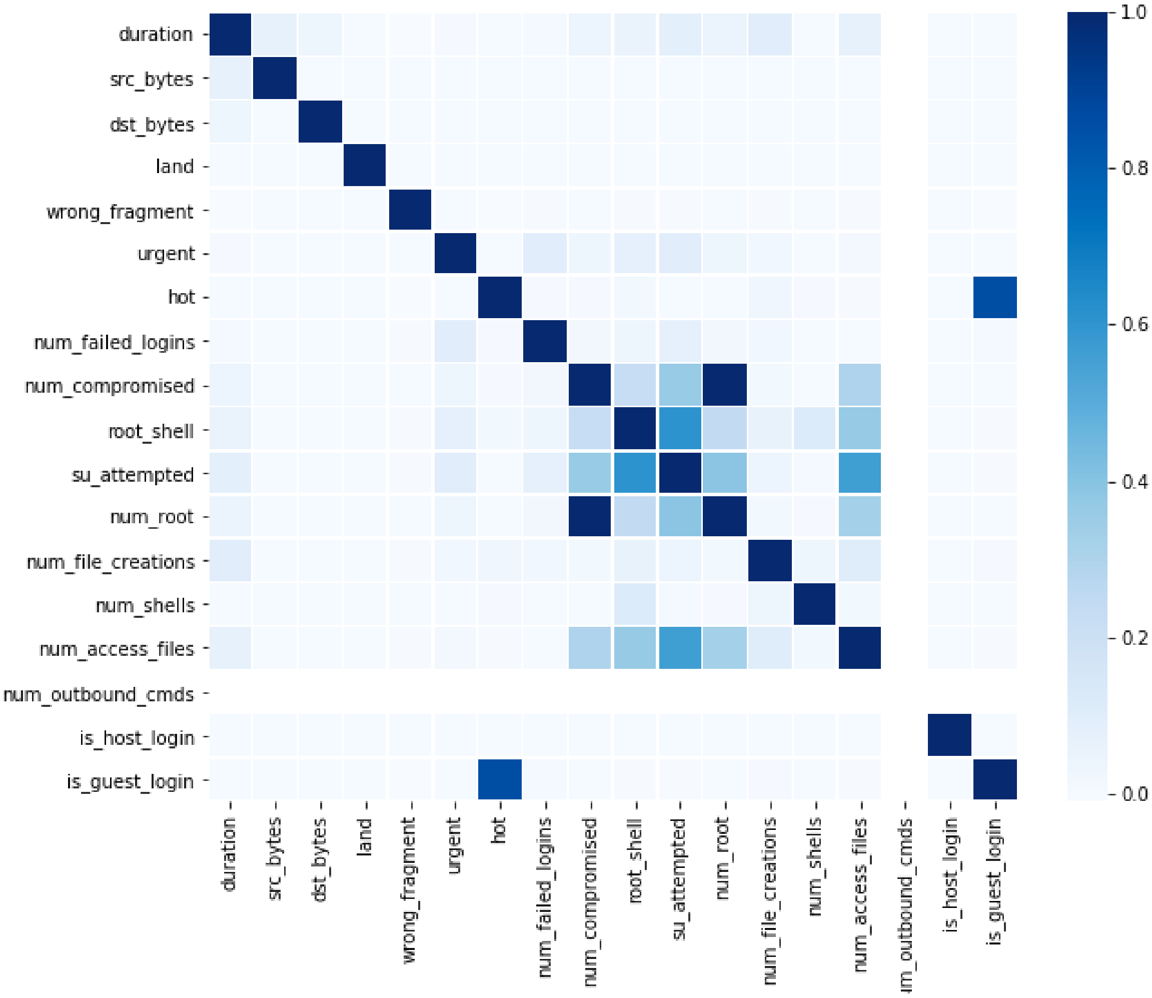


Figure 2: Cross-correlation of the *not_required* attributes

Table 1: Number of categories attained by non-numeric attributes

Attribute	**Number of categories (non-numeric) attained throughout the data-frame**	
Protocol	3	Total = 84 categories
service	70	
flag	11	
connection_type	23	*connection_type* is the label

The *connection_type* attribute is the label of the connection entry/ row in the table. Hence, it is treated as an output rather than a feature for training.

- The remaining 84 categories were each converted as new attribute/column in the dataframe. For example, *protocol* took 3 categories namely, ‘tcp’, ‘udp’ and ‘icmp’. These were transformed in to new attributes – *protocol_tcp*, *protocol_udp*, *protocol_icmp.*

- The newly derived 84 attributes were clubbed with original 38 numeric-valued attributes to yield 2 pre-processed data-frames having 122 features and *connection_type* attribute per sample, for training and testing. In summary, we arrive at two 123-column dataframes with last column bearing the labels.

- The *connection_type* attribute takes up one of 23 distinct non-numerical-valued categories, which can be broadly divided into 5 connection types namely, *normal & anomaly.* They were mapped to numeric values of {0,1} as depicted below:

  normal’ : 0, ’neptune’ : 1 ,’back’: 1, ’land’: 1, ’pod’: 1, ’smurf’: 1, ’teardrop’: 1,’mailbomb’: 1, ’apache2’: 1, ’processtable’: 1, ’udpstorm’: 1, ’worm’: 1, ’ipsweep’ : 1,’nmap’ : 1,’portsweep’ : 1,’satan’ : 1,’mscan’ : 1,’saint’ : 1,’ftp_write’: 1,’guess_passwd’: 1,’imap’: 1,’multihop’: 1,’phf’: 1,’spy’: 1,’warezclient’: 1,’warezmaster’: 1,’sendmail’: 1,’named’: 1,’snmpgetattack’: 1,’snmpguess’: 1,’xlock’: 1,’xsnoop’: 1,’httptunnel’: 1, ’buffer_overflow’: 1,’loadmodule’: 1,’perl’: 1,’rootkit’: 1,’ps’: 1,’sqlattack’: 1,’xterm’: 1

- The *connection type* attribute transformed to take up numeric values is added back to the pre-processed data-frames.

The net result of the pre-processing is entirely numeric valued training and testing data-frames having 123 columns, with 122 features (which is vector x) and the last column depicting the label **y**.(for approach 2 – the final research of this paper)

For approach 1 : (Which is not considered for the final thesis) Below were the steps performed :

Table 2: Labels assigned to different connection categories

Connection Category	**Label assigned**
Normal	0
Denial of Service attack (DoS)	1
Probe attack	2
R2L attack	3
U2R attack	4

The above labels are assigned only for approach 1 (which is further not considered for approach 2) Details of this approach are further mentioned in implementation and evaluation section for approach 1.

3.3.1 Feature Scaling

The 122 columns bearing values of different attributes span varying ranges where some attributes range between thousands and others infractions. The performance of the learning machine is highly dependent upon the span of values its input feature takes. Hence, it is important to perform feature scaling to bring features to a common range of values.(Pervez and Farid; 2014) The 122-dimensional feature vectors were scaled using Z-score normalization to have zero mean and unity variance across the subsets, using the formula as below:

$$featureValue_{new} = \frac{featurevalue - mean}{standard\,deviation} \tag{1}$$

3.4 Optimal Feature Selection

Finding the optimal number of features for a given learning task is an important machine learning problem. It is important to reduce the size of the feature vector by discarding the features which are irrelevant to learning or are least important. This directly impacts the performance of the machine learning classifier. There are various techniques to select the desired number of features from a set by scoring the features using statistics or analyzing feature-importance using random classifiers, etc. But, to arrive at a number which is optimal for the problem is difficult.

3.4.1 Genetic Algorithm

A genetic algorithm (GA) can be used to find the optimal number of features to be preserved from a feature vector. (Su and Liu; 2017) GA is inspired by the biological genetic sequencing happening over generations. The algorithm starts with an initial population which is a set of approximate solutions. Then, it performs mating between the members of the population, which is performed as a crossover combination computationally. The offspring solutions are subjected to random mutation. This process is repeated for generations while preserving only the fittest offsprings which yield the best performance. Hence, GA arrives at an optimal solution after a certain number of generations. The optimal number of features obtained using GA has been designated as K in the remaining text.

3.4.2 Recursive feature elimination

(Dey and Rahman; 2018) The recursive feature elimination (RFE) algorithm aims at preserving the user-defined number of optimal features from a feature-set. This is achieved by doing successive elimination of least important features by training several classifiers upon the dataset. Here, Random Forest (RF) classifiers are successively fit using the feature vectors to keep on eliminating the weak features.

3.4.3 Select-K-Best

(Zhu et al.; 2018) The Select-K-Best feature selector relies upon the data statistics to select the best K features from a given number of features. Here, the ANOVA F-values, between the feature vectors is used to select the top-scoring K features.

3.4.4 Artificial Neural Network

Neural network inspires systems like ANN. Based on the available data classification systems of ANN "learn" to do a prediction for output. A backpropagation algorithm is one of the regular methods used in ANN. Back propagations compensate each error identified in learning through adjustment of connection weights. A directed cycle between the units is not formed due to the feed-forward neural network as the movement of information occurs only in one direction. From the input node, it forwards to the output nodes through hidden layers. Hence, there are no loops or cycles in the network. There are three layers in the neural network:

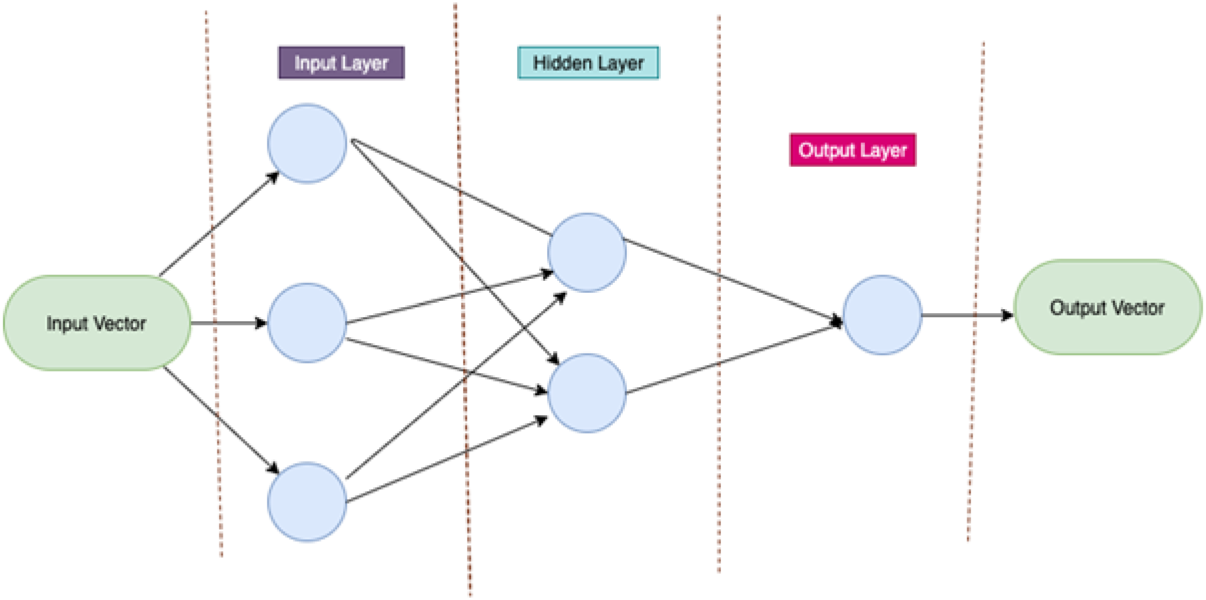


Figure 3: Feed forward neural network

(A) Input Layer: Initial data is contained in the input layer.

(B) Hidden Layer: Hidden layer is the layer between the input layer and the output layer. It can be understood as the middle layer where all the computation occurs.

(C) Output Layer: The output is provided by the output layer after processing of input data. The processing of input data occurs through the hidden layer.

3.4.5 Back propagation algorithm

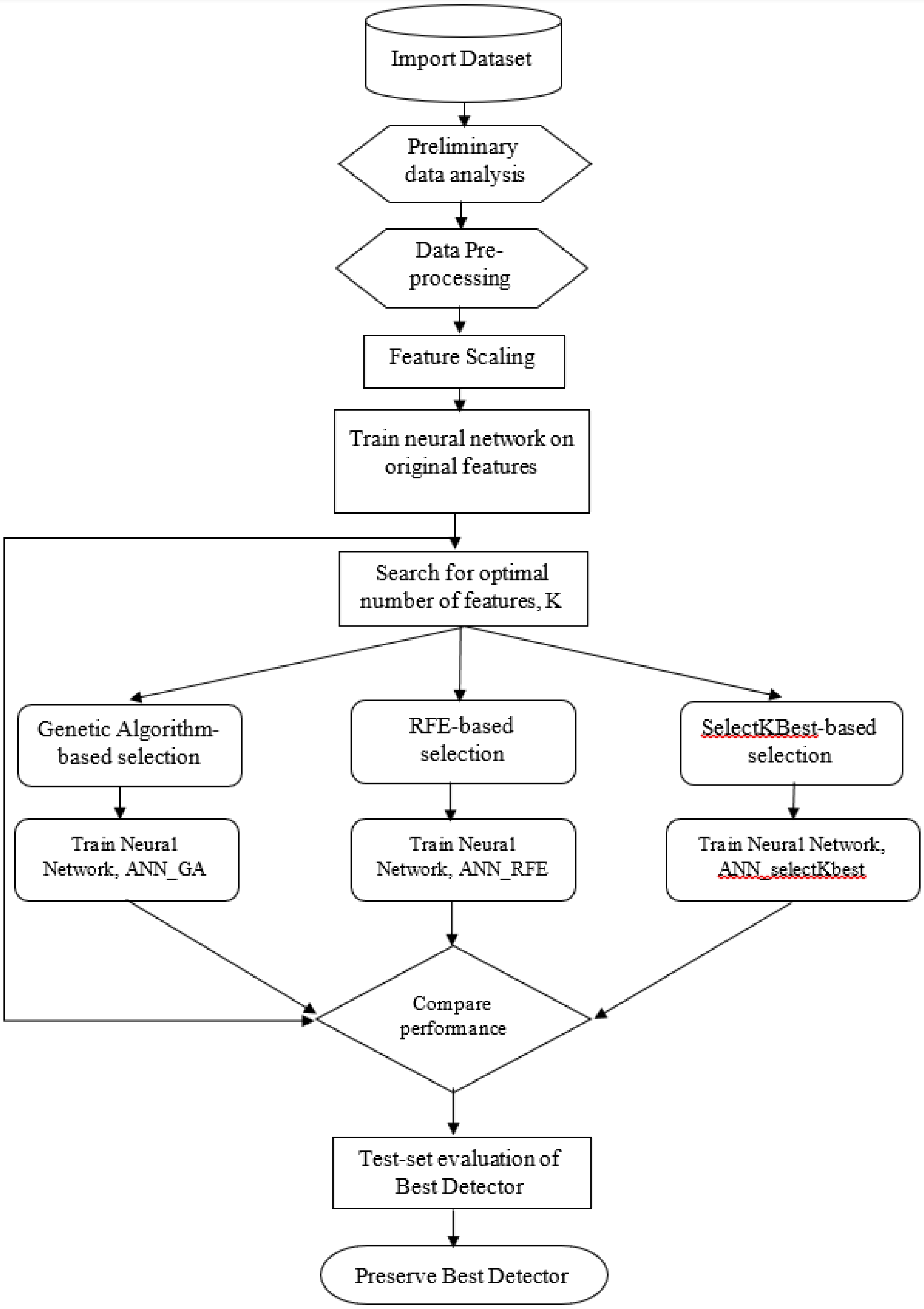
The backpropagation algorithm is a supervised learning algorithm. It is divided into two phases.

(i) Propagation

(ii) Weight update

Both phases run in a loop until the model performs well. The output values in the back-propagation algorithm are compared to the target output. The comparison is done for the prediction of the value of the predefined error function. Through the use of loop techniques iteration of error back to the network is done. The algorithm adjusts the weights of connections depending upon the information received from back feeding. It also reduces the error function's value by a minute amount. When the process is iterated for a huge quantity of loops, a reduction in the value of the error function occurs to assure that a target function is learned by the model. In a way, backpropagation can be understood as a gradient descent technique or method to reduce overall squared error provided by the output which is calculated through the model. In ANN, in case of systems of intrusion detection, self- learning paradigm is used as it has a situation of fixed input and fixed output. On the basis of classification created by the random forest algorithm, automatically ANN starts working on the dataset, and predictions for intrusion detection is done. Instead of statistical models, the neural network is used because statistical models generate output based on pre-assumed behaviors, however, in the neural network, assumptions are a significantly less and continuous modification of assumption can be observed throughout the learning process which leads to a consequential reduction in rates of false negative and false positives.

4 Design Specification



Import Dataset
Preliminary data analysis
Data Pre-processing
Feature Scaling
Train neural network on original features
Search for optimal number of features, K
Genetic Algorithm-based selection
RFE-based selection
SelectKBest-based selection
Train Neural Network, ANN_GA
Train Neural Network, ANN_RFE
Train Neural Network, ANN_selectKbest
Compare performance
Test-set evaluation of Best Detector
Preserve Best Detector

Figure 4: Flowchart of the implementation

5 Implementation

5.1 Machine Learning

(Karimi et al.; 2019) Due to the robustness and adaptability offered by the artificial neural network, we experimented with neural networks for preparing classifiers for predicting the connection type, whether *normal* or *anomaly*.

5.2 Neural Network Architecture

(Zaki et al.; 2019) A shallow neural network with 2 hidden neural layers was designed to accept the training inputs and emit the prediction probability of finding anomaly. The table below summarizes its architecture. Sigmoid activation is used in the output layer. The sigmoid output ranges in floating-point values between 0 and 1, reflecting the confidence of the detection of an anomaly. (Isaac et al.; 2018) Rectified Linear Units (ReLU) activation has been used to normalize the feature maps. (Pomerat et al.; 2019) The activation function is computationally cheaper, which simply truncates the negative values to zero while leaving non-negative values unchanged.

Table 3: Layer-wise description of deployed neural network

Layer Number	**Type of layer**	**Output Shape**
0	Input	(None, *feature_vector_length*)
1	Dense with ReLU	(None, 256)
2	Dropout (20%)	(None, 256)
3	Dense with ReLU	(None, 32)
4	Dropout (20%)	(None, 32)
5	Dense with ReLU	(None, 1)

5.3 Training details

In all trials, the neural network was trained upon the training samples while monitoring validation loss at the end of each epoch for validation. Adam optimizer is chosen due to its computational efficiency and ease of use- requiring negligible tweaking of hyper-parameters (Majdani et al.; 2018).

Table 4: Learning details of neural network

Parameter/Algorithm	**Detail**
Batch Size	32 samples
Number of Epochs	100
Shuffle training data	Yes
Adam Optimizer	learning_rate=0.001, beta_1=0.9, beta_2=0.999, epsilon=1e-07
Loss Function	Binary Cross-entropy
Regularization Technique	Early stopping while expecting decrement in validation loss with patience of 10 epochs

The resultant classifier is evaluated upon testing-subset to find the precision, recall, and F1-scores, and plot confusion matrix. The receiver operating characteristic (ROC) curve and the area under ROC (AUROC) are computed for each classifier.

5.4 Modelling techniques for approach 1 :

(Approach 2 is the final thesis presented in this paper, below are just the highlights of modeling techniques applied for approach 1)

5.4.1 Modelling Technique 1

The experiment aimed to train the neural network upon the arrived 122-dimensional feature vectors and associated 5-class labels.

Neural Network Architecture

A shallow neural network with 3 hidden neural layers was designed to accept the training input and emit the categorical confidence probabilities of the respective 5 classes. The table below summarizes its architecture. Softmax activation is used in the output layer. Rectified Linear Units (ReLU) activation has been used to normalize the feature maps.

Training details

The neural network was trained upon the training subset while monitoring testing loss at the end of each epoch for validation. Adam optimizer is chosen due to its computational efficiency and ease of use- requiring negligible tweaking of hyper-parameters. The resultant classifier is evaluated upon testing-subset to find the precision, recall, and F1-scores, and plot confusion matrix.

5.4.2 Modelling technique 2

This experiment aimed at using reduced-dimension feature vectors for efficient learning by the subsequent neural network.

Feature selection

We used Recursive Feature Elimination (RFE) which uses a chosen ML model to train upon all the possible feature subsets to ultimately remove all the weak features until a specified number of features are left in hand. Here, we deployed Random Forest-based RFE for ranking the 122 features. Random forest works by training several classification trees upon the random subset of samples and outputs prediction by taking the majority vote of these trees on test data. RFE yielded the best 20 features, which were subsequently standardized to zero mean and unit variance.

Neural Network Architecture

It was again a 3-hidden layer neural network accepting 20-dimensional input features. The architecture is similar to the previous ANN, with the addition of an L2 regularizer (at each neural layer) for limiting the L-2 norm of the neuron weights for preventing overfitting. Softmax activation is used in the output layer.

5.4.3 Modelling technique 3

This experiment aimed at preparing 4 binary classifiers each for DoS, Probe, U2R, and R2L attacks by training neural networks upon the 4 child data frames with prior feature selection.

Feature selection

We used SelectKBest algorithm based on Chi-squared statistics of the feature sets, to arrive at the best 15 features from each of the 4 child data frames.

Neural Network Architecture

The architectures for the 4 binary classifiers to detect 4 different types of attacks are identical, which accept 15-D feature vectors to output decision whether it is that type of attack or not. Note, that the y labels are one-hot encoded for training binary classifiers with architecture as below.

Table 5: Outcome for approach1

Connection Category	**Label assigned**
ANN using 122 features	69.36%
ANN using RFE	45.66%
ANN using 27 features (Univariate selection)	70.92%
Ensemble (One vs All)	75.52%

As per our observation, we can see from the above set of experiments that for a given network with defined hidden layers, neurons, parameters, and hyperparameters the prediction accuracy varied based on the feature selection approach.

Ensembling as well didn't show much rise in the accuracy. A high value of accuracy on train data reflects that the model is overfitting. This overfitting could be managed by tuning the hyperparameters. The analysis so far has helped us understand how different approaches of feature selection or stacking impacts the accuracy of a deep learning model. So, further moved with approach 2 to find the nearby best detector using a novel approach of genetic algorithm with ANN on the same dataset (NSL-KDD Dataset).

6 Evaluation (Based on approach 2)

6.1 Case 2: Optimal number of features

We explored for the optimal number of features using Genetic Algorithm. We aimed to find a binary mask with a length of 122 where bit at a location of preserving a feature is set to '1' else it is reset to '0'. The algorithm is initialized with a number of such binary masks decided randomly. These parent feature masks of a generation are subjected to crossover and mutation operations to generate new offsprings.(Yan; 2016) The performance of the selected solution is analysed by cross-validating Support Vector Machine (SVM) classifier upon 20% of the training subset. The best performing (fittest) solution was preserved to be parent in next generation. The table below details the hyper-parameters of the GA search.

Figure below shows the learning curve of the GA upon 100 iterations of learning upon 12598 training samples and 12598 testing samples, with no intersection between training and testing samples. The abscissa shows the iteration number and the ordinate represents the value of fitness function which is the accuracy of SVM upon testing samples at each generation.

Table 6: Implementation details of Genetic Algorithm

Hyper-parameter / Method	**Value**
Number of solutions per population	8
Number of mating parents	4
Number of bits under mutation	3
Number of generations	100
Fitness function	Accuracy of the SVM classifier
Training data	10% of the pre-processed training subset
Testing data	10% of the pre-processed training subset

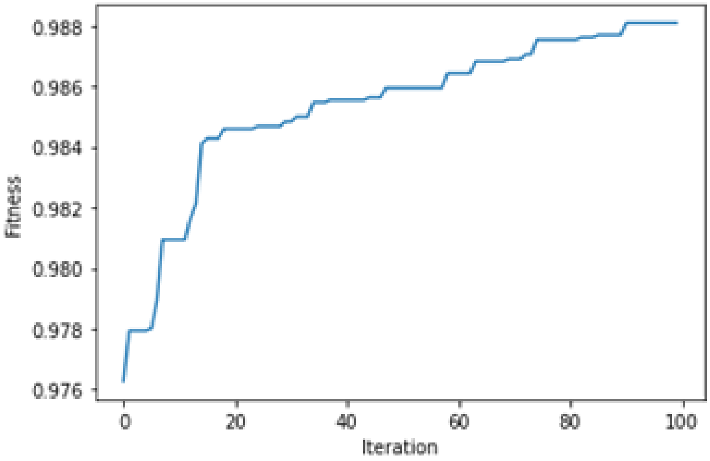


Figure 5: Learning curve of the Genetic Algorithm

The fitness function gradually increments over 100 generations to reach an accuracy of around 98.8%. The number of features selected in the optimal solution is K = 70.

6.2 Network Anomaly Detection

The experiment consisted of total four trials where a feed-forward neural network was trained upon pre-processed features from the training subset of the NSL-KDD dataset. Table below summarizes details of the trials.

For all the trials, a shallow neural network with architecture as depicted in previous chapter was trained in 10-fold cross validation setting upon the respectively supplied feature vectors. The network was trained for 10 epochs upon each data fold. The resulting ac curacies are depicted in table below.

Table 7: Summary of trials with different feature-sets

Case Study #	**Feature selection algorithm**	**Machine learning Model**	**Feature-vector length**	**Detector name**
1	None	Feed-forward Neural Network	122	ANN
2	Genetic Algorithm (GA)	Feed-forward Neural Network	70	ANN_GA
3	Recursive Feature Elimination (RFE)	Feed-forward Neural Network	70	ANN_RFE
4	Select-K-Best	Feed-forward Neural Network	70	ANN_selectKbest
Number of features selected = K = 70				

Further in each trial, the network was retrained upon 80:20 split of training data for 100 epochs with early stopping patience of 10 epochs while expecting decrement in validation loss. The performance trends of the model accuracy and loss are depicted below for all four detectors.

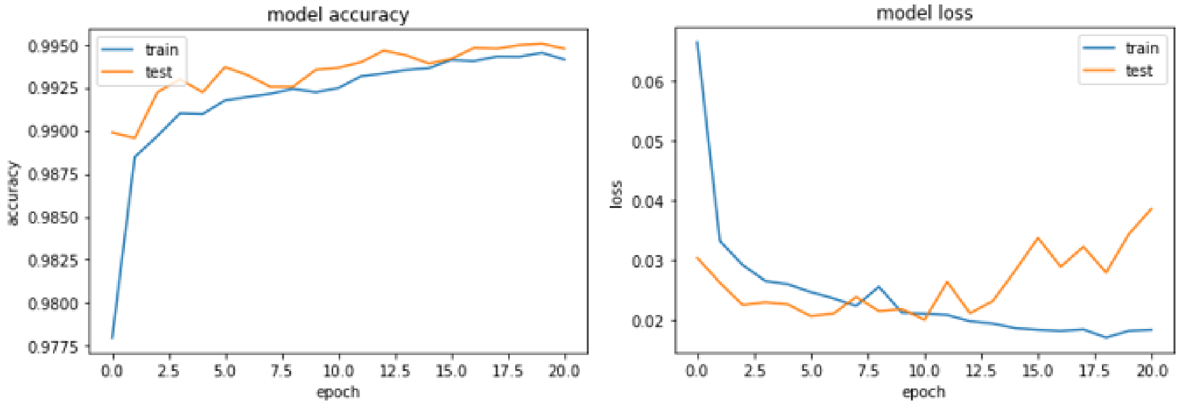


Figure 6: Performance trend of ANN anomaly detector upon 80:20 split of training data

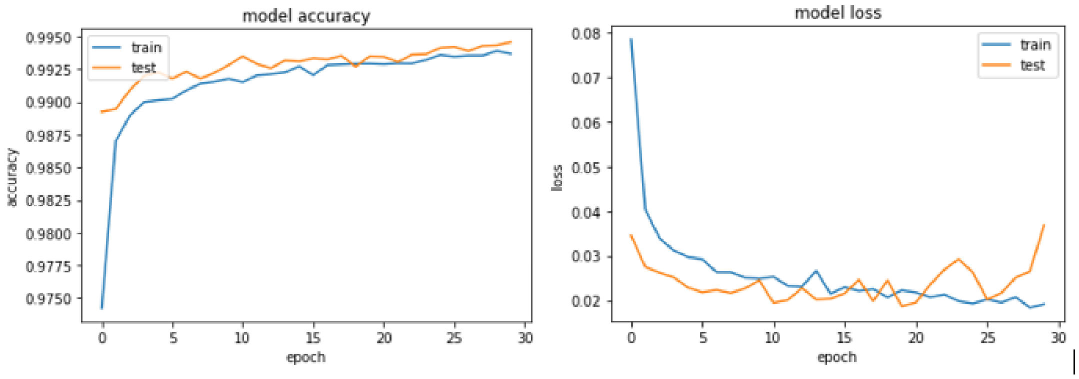


Figure 7: Performance trend of ANN_GA anomaly detector upon 80:20 split of training data

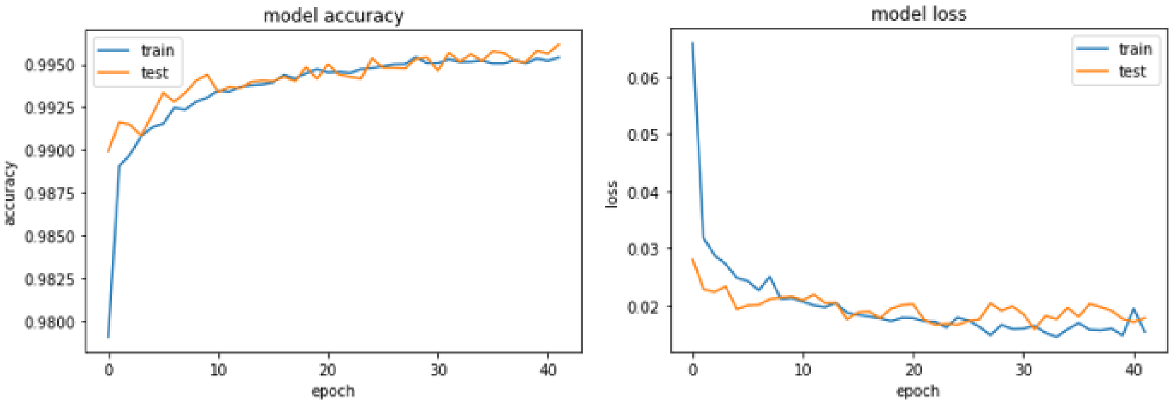


Figure 8: Performance trend of ANN_RFE anomaly detector upon 80:20 split of training data

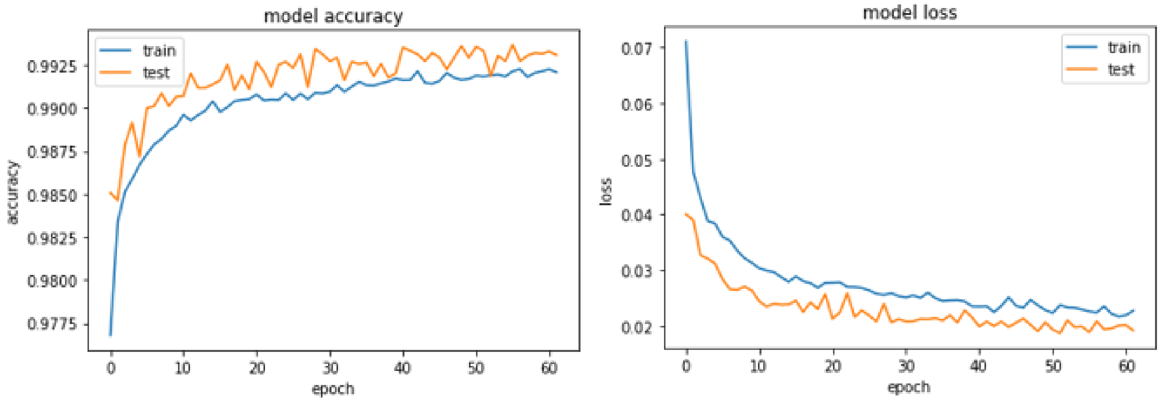


Figure 9: Performance trend of ANN_selectKbest anomaly detector upon 80:20 split of training data

Hence trained 4 detectors were evaluated upon the testing set of the NSL-KDD dataset to note down the testing accuracies. Further, the confusion matrix was obtained for every detector as appended below.

Table 8: Confusion matrix after evaluation of the ANN on test-set

	Normal	**Anomaly**
Normal	8724	987
Anomaly	4195	8638

Further, the Receiver Operating Characteristic (ROC) curves were plotted to compute the area under the ROC (AUROC) in Figure 10.

Table 9: Confusion matrix after evaluation of the ANN_GA on test-set

	Normal	**Anomaly**
Normal	9350	361
Anomaly	4493	8340

Table 10: Confusion matrix after evaluation of the ANN_RFE on test-set

	Normal	**Anomaly**
Normal	8896	815
Anomaly	4168	8665

Table 11: Confusion matrix after evaluation of the ANN_selectKbest on test-set

	Normal	**Anomaly**
Normal	9070	641
Anomaly	4545	8288

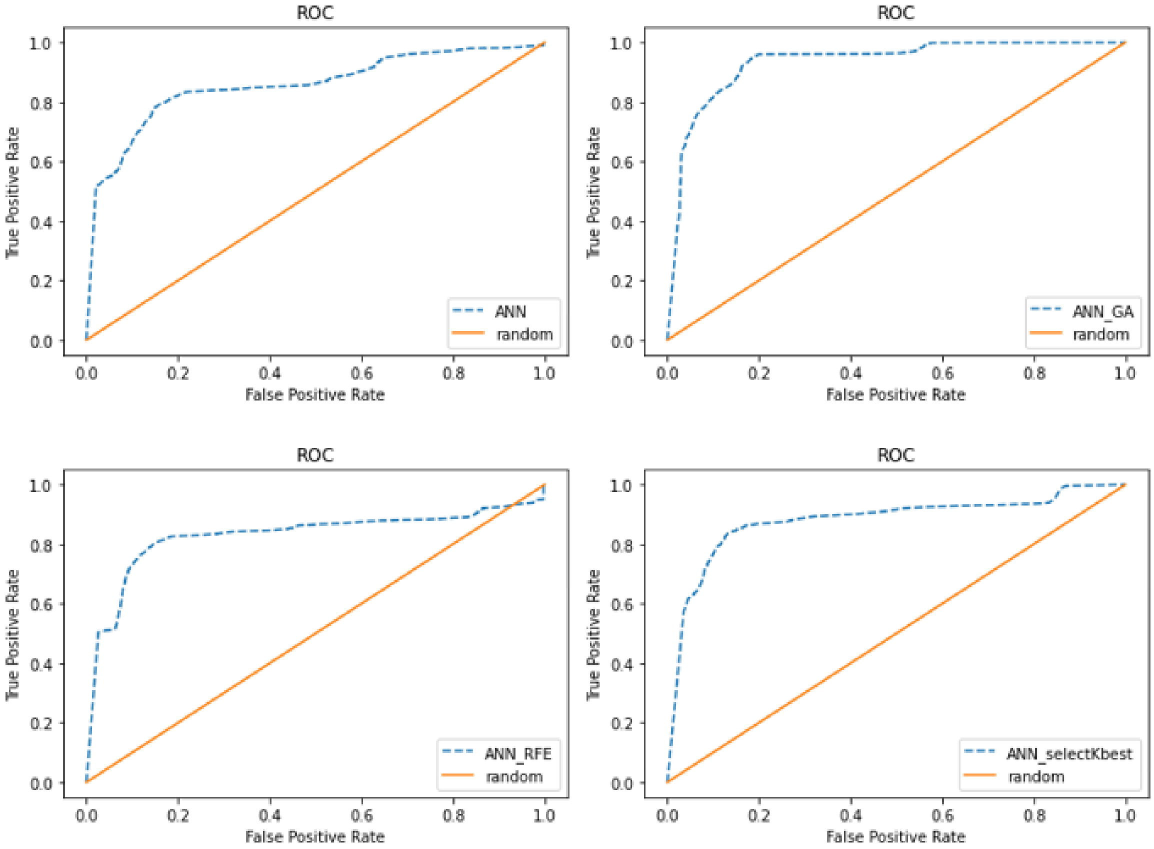


Figure 10: ROC curve for network anomaly detectors in 4 trials

6.3 Comparison of anomaly detectors

Table below shows the performance comparison between the detectors obtained in 4 trials as depicted above.

Table 12: 10-cross fold performance of Anomaly Detectors

Anomaly Detector	**Mean 10-fold CV accuracy(%)**	**Held-out Test -set accuracy(%)**	**Testing F1 -score**	**Area under ROC (AUROC)**
ANN	99.36	77.01	0.770	0.856
ANN_GA	99.32	78.47	0.783	0.934
ANN_RFE	99.35	77.90	0.779	0.829
ANN_selectKbest	99.10	77.00	0.769	0.879

Out of the four trained detectors, the best was obtained using GA-selected features considering the overall F1-score and AUROC upon the testing set. The performance of ANN_GA is highlighted in italicised font.

6.4 Best Anomaly Detector

The ANN_GA was chosen as the final classifier which was trained upon the entire training set of NSL-KDD and actively monitoring its performance upon testing set for validation after each epoch. The test-set accuracy obtained was equal to 79.48%.

Table 13: Confusion matrix after evaluation of the best Anomaly Detector (ANN_GA) on test-set

	Normal	**Anomaly**
Normal	9350	361
Anomaly	4264	8569

Table 14: Classification report of the best performing Anomaly Detector (ANN_GA)

Class	**Precision**	**Recall**	**F1-score**	**No. of samples**
Normal	0.6868	0.9628	0.8017	9711
Anomaly	0.9596	0.6677	0.7875	12833
Weighted Average	0.8421	0.7948	0.7936	22544

Figure below shows the ROC curve of the best detector. The area under the ROC curve was found to be 0.947.

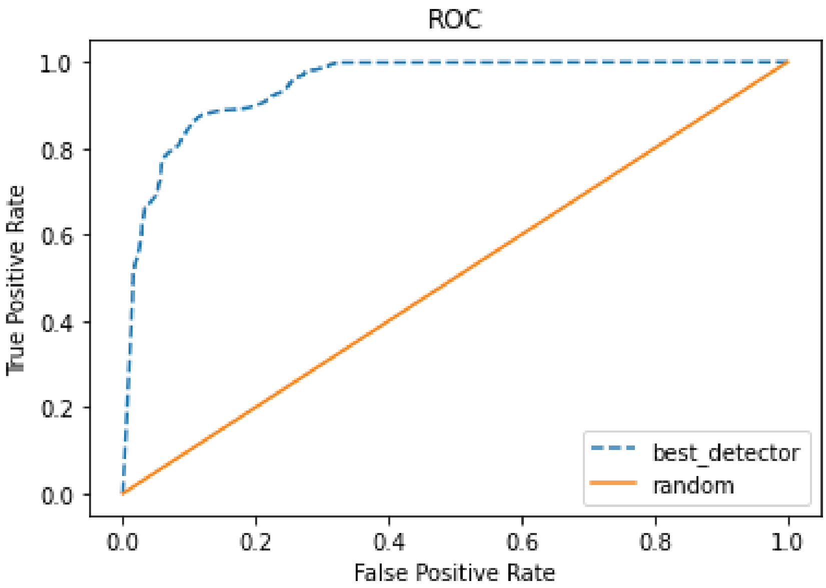


Figure 11: ROC curve for the best Anomaly Detector (ANN_GA)

7 Conclusion and Future Work

The presented work addressed the problem of finding the optimal number of features during the feature selection process, which is an essential step before training machine learning classifiers. This problem was addressed using Genetic Algorithm heuristics which attempts to find the optimal solution iteratively over a certain number of generations while starting with an initial guess.

The GA selects a certain number, K, of features as optimal feature-set. Training a shallow neural network upon the optimal features help achieve better classifier than the classifiers obtained using the same (K) number of features selected using other feature-selection algorithms namely, RFE and SelectKBest algorithms. Further, the classifier has better mean 10-fold cross-validation accuracy upon training subset of public NSL-KDD dataset, than most of the previous methods studied here.

This work studied the effect of three feature selection techniques namely, genetic algorithm, recursive feature elimination, and Select-K-Best algorithm upon the performance of network anomaly detection classifier. The mean 10-fold cross-validation accuracy obtained using ANN_selectKbest detector, which performed the best, is 99.32%. It is higher than the studied previously reported classifiers for network anomaly detection.

More analysis can be performed to achieve the minimum optimal number of features to gain better accuracy for the performance of the model in the future.

References

Behera, S., Pradhan, A. and Dash, R. (2018). Deep neural network architecture for anomaly based intrusion detection system, *2018 5th International Conference on Signal Processing and Integrated Networks (SPIN)*, pp. 270–274.

Bitaab, M. and Hashemi, S. (2017). Hybrid intrusion detection: Combining decision tree and gaussian mixture model, *2017 14th International ISC (Iranian Society of Cryptology) Conference on Information Security and Cryptology (ISCISC)*, pp. 8–12.

Cahyo, A. N., Hidayat, R. and Adhipta, D. (2016). Performance comparison of intrusion detection system based anomaly detection using artificial neural network and support vector machine, *AIP Conference Proceedings* **1755**(1): 070011.

Chaudhary, A., Mittal, H. and Arora, A. (2019). Anomaly detection using graph neural networks, *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, pp. 346–350.

Chitrakar, R. and Huang, C. (2012). Anomaly based intrusion detection using hybrid learning approach of combining k-medoids clustering and naïve bayes classification, *2012 8th International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 1–5.

Dey, S. K. and Rahman, M. M. (2018). Flow based anomaly detection in software defined networking: A deep learning approach with feature selection method, *2018 4th International Conference on Electrical Engineering and Information Communication Technology (iCEEiCT)*, pp. 630–635.

Isaac, B. J., Kinjo, H., Nakazono, K. and Oshiro, N. (2018). Suitable activity function of neural networks for data enlargement, *2018 18th International Conference on Control, Automation and Systems (ICCAS)*, pp. 392–397.

Jones, C. B., Carter, C. and Thomas, Z. (2018). Intrusion detection response using an unsupervised artificial neural network on a single board computer for building control resilience, *2018 Resilience Week (RWS)*, pp. 31–37.

Karimi, M., Jahanshahi, A., Mazloumi, A. and Sabzi, H. Z. (2019). Border gateway protocol anomaly detection using neural network, *2019 IEEE International Conference on Big Data (Big Data)*, pp. 6092–6094.

Kromkowski, P., Li, S., Zhao, W., Abraham, B., Osborne, A. and Brown, D. E. (2019). Evaluating statistical models for network traffic anomaly detection, *2019 Systems and Information Engineering Design Symposium (SIEDS)*, pp. 1–6.

Majdani, F., Petrovski, A. and Petrovski, S. (2018). Generic application of deep learning framework for real-time engineering data analysis, *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8.

Meena, G. and Choudhary, R. R. (2017). A review paper on ids classification using kdd 99 and nsl kdd dataset in weka, *2017 International Conference on Computer, Communications and Electronics (Comptelix)*, pp. 553–558.

Naseer, S., Saleem, Y., Khalid, S., Bashir, M. K., Han, J., Iqbal, M. M. and Han, K. (2018). Enhanced network anomaly detection based on deep neural networks, *IEEE Access* **6**: 48231–48246.

Pervez, M. S. and Farid, D. M. (2014). Feature selection and intrusion classification in nsl-kdd cup 99 dataset employing svms, *The 8th International Conference on Software, Knowledge, Information Management and Applications (SKIMA 2014)*, pp. 1–6.

Pomerat, J., Segev, A. and Datta, R. (2019). On neural network activation functions and optimizers in relation to polynomial regression, *2019 IEEE International Conference on Big Data (Big Data)*, pp. 6183–6185.

Primartha, R. and Tama, B. A. (2017). Anomaly detection using random forest: A performance revisited, *2017 International Conference on Data and Software Engineering (ICoDSE)*, pp. 1–6.

Punitha, A., Vinodha, S., Karthika, R. and Deepika, R. (2019). A feature reduction intrusion detection system using genetic algorithm, *2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)*, pp. 1–7.

Sahu, N. K. and Mukherjee, I. (2020). Machine learning based anomaly detection for iot network: (anomaly detection in iot network), *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184)*, pp. 787–794.

Sani, Y., Mohamedou, A., Ali, K., Farjamfar, A., Azman, M. and Shamsuddin, S. (2009). An overview of neural networks use in anomaly intrusion detection systems, *2009 IEEE Student Conference on Research and Development (SCOReD)*, pp. 89–92.

Shah, B. and Trivedi, B. H. (2015). Reducing features of kdd cup 1999 dataset for anomaly detection using back propagation neural network, *2015 Fifth International Conference on Advanced Computing Communication Technologies*, pp. 247–251.

Su, Q. and Liu, J. (2017). A network anomaly detection method based on genetic algorithm, *2017 4th International Conference on Systems and Informatics (ICSAI)*, pp. 1029–1034.

Subba, B., Biswas, S. and Karmakar, S. (2016). A neural network based system for intrusion detection and attack classification, *2016 Twenty Second National Conference on Communication (NCC)*, pp. 1–6.

Yan, G. (2016). Network anomaly traffic detection method based on support vector machine, *2016 International Conference on Smart City and Systems Engineering (ICSCSE)*, pp. 3–6.

Zaki, P. W., Hashem, A. M., Fahim, E. A., Masnour, M. A., ElGenk, S. M., Mashaly, M. and Ismail, S. M. (2019). A novel sigmoid function approximation suitable for neural networks on fpga, *2019 15th International Computer Engineering Conference (ICENCO)*, pp. 95–99.

Zhang, Y., Yang, Q., Lambotharan, S., Kyriakopoulos, K., Ghafir, I. and AsSadhan, B. (2019). Anomaly-based network intrusion detection using svm, *2019 11th International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1–6.

Zhu, Z., Cheng, R., Do, L., Huang, Z. and Zhang, H. (2018). Evaluating top-k meta path queries on large heterogeneous information networks, *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 1470–1475.