Turkish Journal of Computer and Mathematics Education *Vol.13 No.02 (2022), 725-733*

Research Article

Intrusion detection System using Random Forest Approach

B.Yogesh[a], Dr. G. Suresh Reddy[b]

[a]VNR VJIET, Pragati Nagar, Nizampet (S.O.), Hyderabad 500090

[b]Ph.D.(CSE)Professor, Department of IT,VNRVJIET,Hyderabad

Corresponding author: B. YOGESH (yogeshace999@gmail.com)

Abstract: The advancing area of technology presents a attack of new attacks for the criminals and the specialists. It presently gotten to be a major concern within the cyberspace. It is the advancement in the computer elevated for the protection of programs by various programmers. The reason these systems display the attacks that are encountered in the internet, by using alternate in the regular attack. The classification algorithms are used for analyzing NSL Dataset with Attributes. The classification used in this project are support SVM, RFC, K Neighbors Classifier, Logistic Regression, Naive bayes. Feature extraction is part of the RFE. These are used to test the attacks on various classes. The results that are found that the RFC gives the performance more compared to the other classification with the high accuracy.

Keywords: Intrusion Detection System (IDS), SVM, NSL dataset, Logistic Regression, NB Classifier, Random Forest Algorithm (RFA).

1.Introduction

The tremendous growth and technical advancement in current times, cybersecurity has become a rising challenge. The internet makes available all of mankind's knowledge, and with the development of mobile computing, it is now at everyone's fingertips, Cybercrime, cyberattacks, become all too common. A statement from the anti-phishing working group, around 227,000 virus detections occur every day which has approximately 20 million new malwares every day. Malwares is a malicious computer software designed to attack a system. In the past, dealing with malware was simple, but with the cybercrimes are increasing over the last two decades, cybersecurity strategies are also evolving into more intelligent approaches.

The biggest issue that has arisen as a result of technological advancements and the network is the necessary to inject on a target machine. Anyone who wants to launch an attack can use various types scripts and advanced programmes to bypass and avoid security measures, and cyber-attacks by unskilled criminals are on the rise. According to an APWAnalysis from 2016, phishing attempts cost billions of dollars, the retail industry has been the target of several of these attacks.

It is in charge of a big volume of informationand is critical in identifying many types of intrusions. Because intrusion detection relies on a good classifier, IDS is classified asa classification challenge. Intrusion is described as a harmful act that compromises a computer system. The anomalies and IDS have become an important part of system defense as a result of network attacks.

2. Significance of The Study

When it comes to the IDS, numerous methods and strategies have been utilized in the past and will be employed in the near future. Since everybody wants a solution to their network security challenges that is 100 percent safe and dependable. Despite substantial research into intrusion detection systems and a plethora of antivirus software, there are no reliable solutions until all of the intrusion detection system's criteria are satisfied, which are time consuming, cumbersome, and require frequent upgrading. In this we will discuss the challenges that IDS developers confront in establishing an effective intrusion detection system, as well as its limitations and difficulties related to its development and organization.

The amount of money spent on building effective and reliable intrusion detection systems is on the rise, as is cybercrime. Cybercrime costs to world $100 billion per year. According to estimates, network attacks affect over 556 million people each year, amounting to 18 victims every second.

3. Review of Related Studies

In order to stay current, intrusion detection systems have traditionally relied substantially on human involvement. This dependency can be seen in the addition of blacklisting newly detectedURLs used by phishers, the addition of systems with a set of rules, the generation of errors, and whitelist filtering, to name a few instances. With the internet's rapid expansion and the vastness of current networks, as well as the high volume of minimal attacks produced, as you can see how a human-based intrusion detection system would fall short.

One method for addressing this problem is ML. First and foremost, learning from computers improves with volume; a predictive model effectively learns from prior breaches and planned intrusions, as well as the behaviors that accompany them and also how they differ in normal activity. Once these patterns have been discovered, a machine learning system recognizes and classify them rapidly and on a larger scale than any techniques.

The usage of labelled training data is required in the Supervised Learning class. The purpose of supervised learning algorithms is prediction. In intrusion detection, the SVM and RFC are two instances of supervised learning. SVM is widely utilized in network-based systems that detect intrusionsbecause of its processing efficiency. Semi-supervised models differ from supervised models in that they may train on unlabeled data, allowing them to recognise patterns and make classifications without the need for human intervention. This capacity comes in handy when vast quantities of tagged training examples are in short supply or missing.

Unsupervised learning is a classification system that is taught on data that has not been labelled. In order to detect intrusions, the unsupervised learning system will attempt to reveal latent structure in unlabeled data. For unsupervised learning, no training data is available. Clustering, association analysis, and dimensionality reduction are all tools that can be used to aid in this process. Examples of clustering approaches include K-Means and C-Means. Both SVD and PCA can be used to reduce dimensionality.

In supervised, data is categorizedand the outcomes are predictable and mappable.That is to say, in both instances, algorithm was constructed using analytics with known outcomes. A classification challenge occurs when the model is asked to categorize a dataset object. If specific conditions are met, attributes are supplied depending on the train dataset's resemblance to a class, What category does an item fall is to be assigned. When a model tries to anticipate what is missing in a dataset, it is called a regression problem. Regression problems are more commonly utilized on continuous data than classification problems.

4. Objectives of The Study

- Theyareusedtodetectfalseerrorsusingvarioustechniquesofthealgorithms.
- To Compare boththeapproachesarecomparedinterms ofaccuracyandsensitivity.
- UsingRandomForestClassifier,checkandvalidatetheoutcomesoftheproposed methodology
- To calculate best accuracy, precision, confusion matrix of classification algorithms.

5. Hypotheses of The Study

- Random forest performed well at detecting abnormalities in the NSL Dataset. This was believed it will be capable of classifying accuracy correctly using the NSL Kdd.
- It should be able to handle a variety of tasks of producing positive outcomes within intrusion detection systems, as well as delivering the best possible solutions to challenges.

- The dataset is clean data only two columns are of nulls host login and num outbound cmds.

6.About the Data-set

NSL data-set

NSL is a data set designed to alleviate a few of the underlying issues in the old version data set. Despite the fact some of the data in this version of the data collection is still missing McHugh's issues and it's possible that this isn't a perfect representation of what's already out therenetwork systems, we assume it could be used as a useful standard set of data for comparing different intrusion detection methods by researchers due to a scarcity of publicly available network-based IDS data sets.

7. Machine learning process

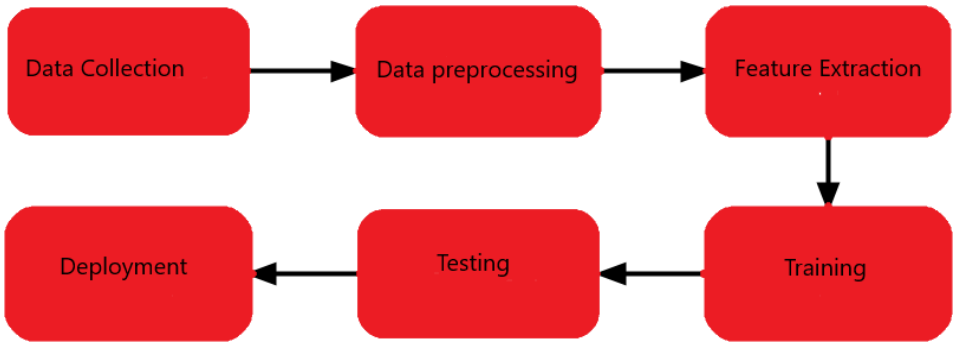


Figure.1 FlowDiagramof ML

7.1 Load Dataset

Depending on the project, appropriate research data is gathered and kept in memory (Chumachenko, 2017).

```
In [2]: train = pd.read_csv(r'E:\New folder\New folder\network_intrusion_detection-main\Train_data.csv')
        test = pd.read_csv(r'E:\New folder\New folder\network_intrusion_detection-main\Test_data.csv')
        train
```

Out[2]:

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_srv_count	dst_host_same_srv_rate	dst_l
0	0	tcp	ftp_data	SF	491	0	0	0	0	0	...	25	0.17	
1	0	udp	other	SF	146	0	0	0	0	0	...	1	0.00	
2	0	tcp	private	S0	0	0	0	0	0	0	...	26	0.10	
3	0	tcp	http	SF	232	8153	0	0	0	0	...	255	1.00	
4	0	tcp	http	SF	199	420	0	0	0	0	...	255	1.00	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
25187	0	tcp	exec	RSTO	0	0	0	0	0	0	...	7	0.03	
25188	0	tcp	ftp_data	SF	334	0	0	0	0	0	...	39	1.00	
25189	0	tcp	private	REJ	0	0	0	0	0	0	...	13	0.05	
25190	0	tcp	nnsp	S0	0	0	0	0	0	0	...	20	0.08	
25191	0	tcp	finger	S0	0	0	0	0	0	0	...	49	0.19	

25192 rows × 42 columns

7.2 Data pre-processing

7.2.1 Scaling numerical Attributes: To standardize characteristics, remove the mean and scale to unit variance.By computing each characteristic is independently focused and modified statistical data about the cases in the training phase The standard error and average are then generated and used to alter the data that follows. Many approximations require dataset uniformity they may perform poorly if indeed the particular attributes do not match standard data which is normally distributed.

7.2.2 Encoding Categorical Attributes: In Python Label Encoding, we need to replace the category value with a numerical value range from 0 to the entire class labels minus one. If the categorical variable's result

comprises six sections, for example, we'll pick 0, 1, 2, 3, 4, and 5.

```
In [14]: #Scaling Numerical Attributes
scaler = StandardScaler()

# extract numerical attributes and scale it to have zero mean and unit variance
cols = train.select_dtypes(include=['float64','int64']).columns
sc_train = scaler.fit_transform(train.select_dtypes(include=['float64','int64']))
sc_test = scaler.fit_transform(test.select_dtypes(include=['float64','int64']))

# turn the result back to a dataframe
sc_traindf = pandas.DataFrame(sc_train, columns = cols)
sc_testdf = pandas.DataFrame(sc_test, columns = cols)
print(sc_traindf.head())
print(sc_testdf.head())
```

```
   duration  src_bytes  dst_bytes     land  wrong_fragment    urgent  \
0 -0.113551  -0.009889  -0.039310 -0.00891       -0.091223 -0.006301
1 -0.113551  -0.010032  -0.039310 -0.00891       -0.091223 -0.006301
2 -0.113551  -0.010093  -0.039310 -0.00891       -0.091223 -0.006301
3 -0.113551  -0.009996   0.052473 -0.00891       -0.091223 -0.006301
4 -0.113551  -0.010010  -0.034582 -0.00891       -0.091223 -0.006301

        hot  num_failed_logins  logged_in  num_compromised  ...  \
0 -0.091933           -0.02622  -0.807626        -0.021873  ...
1 -0.091933           -0.02622  -0.807626        -0.021873  ...
2 -0.091933           -0.02622  -0.807626        -0.021873  ...
3 -0.091933           -0.02622   1.238197        -0.021873  ...
4 -0.091933           -0.02622   1.238197        -0.021873  ...
```

Figure.2 Scaling numerical attributes

```
In [15]: #Encoding Categorical Attributes
encoder = LabelEncoder()

cattrain = train.select_dtypes(include=['object']).copy()
cattest = test.select_dtypes(include=['object']).copy()

traincat = cattrain.apply(encoder.fit_transform)
testcat = cattest.apply(encoder.fit_transform)

enctrain = traincat.drop(['class'], axis=1)
cat_Ytrain = traincat[['class']].copy()
print(traincat.head())
print(testcat.head())
```

```
   protocol_type  service  flag  class
0              1       19     9      1
1              2       41     9      1
2              1       46     5      0
3              1       22     9      1
4              1       22     9      1
   protocol_type  service  flag
0              1       45     1
1              1       45     1
2              1       19     9
3              0       13     9
4              1       55     2
```

Figure.3 Encoding categorical attributes

7.2.3 Feature Extraction In a statistical model, this is the process of decreasing the number of variables. To reduce the total cost of computing and, in certain situations, to increase the performance of the model, the number of variables should be minimized. Statistics are used to evaluate the relationship between each input parameter and the goal parameter, and the input parameters having the strongest association to the target attribute are chosen. Although the type of data in both the dependent and independent variables influences the statistical measures used, these procedures can be quick and effective.Rather than relying on a single decision tree, the random forest gathers estimate from every tree and estimates the correct outcome based on the amount of votes. It takes lesser time to train compared to other techniques. It accurately predicts outcome and runs rapidly, even with a big dataset. It can maintain accuracy even if a significant amount of data is absent.

7.3 SplittingtheDatasetintotheTrainandTestsets

In this, we partition our sample into a train set and a test set. Because it increases efficiency of our classification model, this is a critical step in data preprocessing. Suppose we've built our classification model on one sample before putting it to the test on another. Our model will therefore struggle to comprehend the links

between both the models. The model's performance will deteriorate if we train it adequately and its training accuracy is good, but then give it a new dataset. As a result, we make every effort to develop a classification model that fits well with both train and test sets. The train set is a part of the sample used to build the classification model, and the result is now known for these samples. The test set is a part of the sample that is used to test the model, and it uses to test set to forecast output.

Figure.4 Feature Selection

```
In [21]: #Dataset Partition
         X_train,X_test,Y_train,Y_test = train_test_split(train_x,train_y,train_size=0.60, random_state=2)
```

Figure.5 Divide the data into two categories train and test.

7.4 Evaluation Metrics

Accuracy. Theterm"accuracy"referstohowwellamodelpredictsagivensetofdata(Shung,2018).

$$Accuracy = \frac{Number\ of\ Correct\ predictions}{Total\ number\ of\ predictions\ made}$$

F1-Score.F1Score isimportantwhentheclassisunevenandusesboththeaccuracyandtherecall.

$$F1 - score = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}}$$

Precision. Thefractionofgenuinepositivesintheoverallnumberofpositivesisknownasprecision.

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

Recall. Theproportionofaccuratelyidentifiedpositivesisknownasrecall(Shung,2018)

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

7.5 Train the models

7.5.1 SVM: Vapnik designed this algorithm in 1979 [12]. The basic SVM, often known as a binary classification problem, deals with two class problems. Its generalisation capacity is greater than those of other traditional learning approaches. A hyperplane is used to separate a number of support vectors in this study.The SVM is one of the most widely used classification algorithm for binary classification. The use of the SVM is justified because it outperforms all other classifiers. Second, the SVM is flexible because it is based on a classlabel that is unaffected by the size of the feature space and is insensitive to the amount of data points. As a result, it is a better method for training a larger set of patterns than neural networks.

```
============================== Support Vector Machine Model Evaluation ==============================

Cross Validation Mean Score:
 0.9621565748882095

Model Accuracy:
 0.9628183923255045

Confusion matrix:
 [[6592  462]
 [ 100 7961]]

Classification report:
              precision    recall  f1-score   support

     anomaly       0.99      0.93      0.96      7054
      normal       0.95      0.99      0.97      8061
```

Figure.6 Accuracy for Support vector machine

7.5.2 Logistic regression: For binary classification issues, logistic regression is a fundamental machine learning approach. In today's world, it's mostly employed to create a basic model. Nonetheless, it's a great first algorithm to create because it's so easy to understand. Logistic regression is comparable to linear regression in certain ways. For creating predictions, we're still using a line equation. To transform real values to probabilities, the results are sent through a Sigmoid activation function this time. The probability indicates the likelihood that the instance belongs to a positive. Based on a threshold value, these probabilities are converted to actual classes. We assign the positive class if the likelihood is greater than the threshold, and vice versa. Depending on the situation and the sort of statistic you're optimising for, the threshold value can (and should) be changed.

```
============================== LogisticRegression Model Evaluation ==============================

Cross Validation Mean Score:
 0.9538203526869973

Model Accuracy:
 0.9544161429043996

Confusion matrix:
 [[6627  427]
 [ 262 7799]]

Classification report:
              precision    recall  f1-score   support

     anomaly       0.96      0.94      0.95      7054
      normal       0.95      0.97      0.96      8061
```

Figure.7 Accuracy for Logistic regression

7.5.3 KNN: The KNN is the fundamental mechanism, as it is based on the Supervised approach. The model implies that the potentially new instance and previous instances are equivalent, and it places the new instance in the class that is the most similar to the previous classes. It saves all data available and categorizes fresh data

points depending on how similar they are to previous data. This indicates that using the K-NN approach, fresh data can be swiftly sorted into classification.

```
============================== KNeighborsClassifier Model Evaluation ==============================

Cross Validation Mean Score:
 0.9910023145959611

Model Accuracy:
 0.9930532583526298

Confusion matrix:
 [[6984   70]
 [  35 8026]]

Classification report:
              precision    recall  f1-score   support

     anomaly       1.00      0.99      0.99      7054
      normal       0.99      1.00      0.99      8061
```

Figure.8 Accuracy for K-Nearest Neighbor

7.5.4 NB Classifier: For multiples and categorization issues, the NB classifier is a probabilistic approach. When confronted with linear or qualitative data, the approach is the easiest to grasp. Since the chances for each assumption are decreased to keep the computation manageable, it is known as NB or silly Bayes.By assuming a Gaussian distribution, which is the most general idea, Naive Bayes can be extended to legitimate characteristics. Gaussian Naive Bayes is a naive Bayes modification. Different algorithms would be used to establish the data distribution, however the Stochastic (or Normal) allocation is the most straightforward to deal with just because allyou have to do now is utilise your train data to obtain the summary statistics.

```
============================== Naive Baye Classifier Model Evaluation ==============================

Cross Validation Mean Score:
 0.9067137727213834

Model Accuracy:
 0.9067813430367185

Confusion matrix:
 [[5981 1073]
 [ 336 7725]]

Classification report:
              precision    recall  f1-score   support

     anomaly       0.95      0.85      0.89      7054
      normal       0.88      0.96      0.92      8061
```

Figure.9 Accuracy for Naive Bayes Classifier

7.5.5 Random Forest: It is built on the basis of a forest of trees using randominputs. Random Forest approachconstructs trees using a separate bootstrap sample of the data, so classification and regression treesareformeddifferently.Innormaltrees,theoptimalsplitamongallvariablesisusedtospliteachnode.When compared to other classifiers such as Neural Networks (NN), discriminant analysis, and soon, this traditional technique performs remarkably well. It also has a strong resistance to dataoverfitting. Thetechniquemaybebiasedtowardvariableswithalargenumberofcategories.It'snoproblemtotryoutthiscorrelatedda tatrickonourdataas longasthis trickisremembered.

```
In [27]: nnb = RandomForestClassifier()
         nnb.fit(X_train, Y_train)
         nnb.score(X_test, Y_test)

Out[27]: 0.9970229234891337

In [28]: y_preds = nnb.predict(X_test)
         y_preds

Out[28]: array(['anomaly', 'anomaly', 'anomaly', ..., 'normal', 'normal', 'normal'],
               dtype=object)

In [29]: from sklearn.metrics import classification_report, confusion_matrix
         cn_m = confusion_matrix(Y_test, y_preds)
         cn_m

Out[29]: array([[4674,   15],
                [  15, 5373]], dtype=int64)
```

Figure.10 Accuracy for Random Forest

```
In [30]: fig, ax = plt.subplots()
         ax = sns.heatmap(cn_m, annot= True, fmt='d', cbar=False,
                          xticklabels=["normal", "anomaly"],
                          yticklabels=["normal", "anomaly"]);

         plt.xlabel("Actual value",
                    fontsize=15)
         plt.ylabel("Predicted value",
                    fontsize = 15);
         plt.title("Confusion matrix",
                   fontsize=20);
```

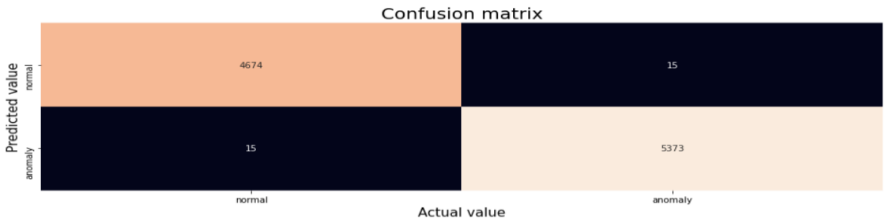
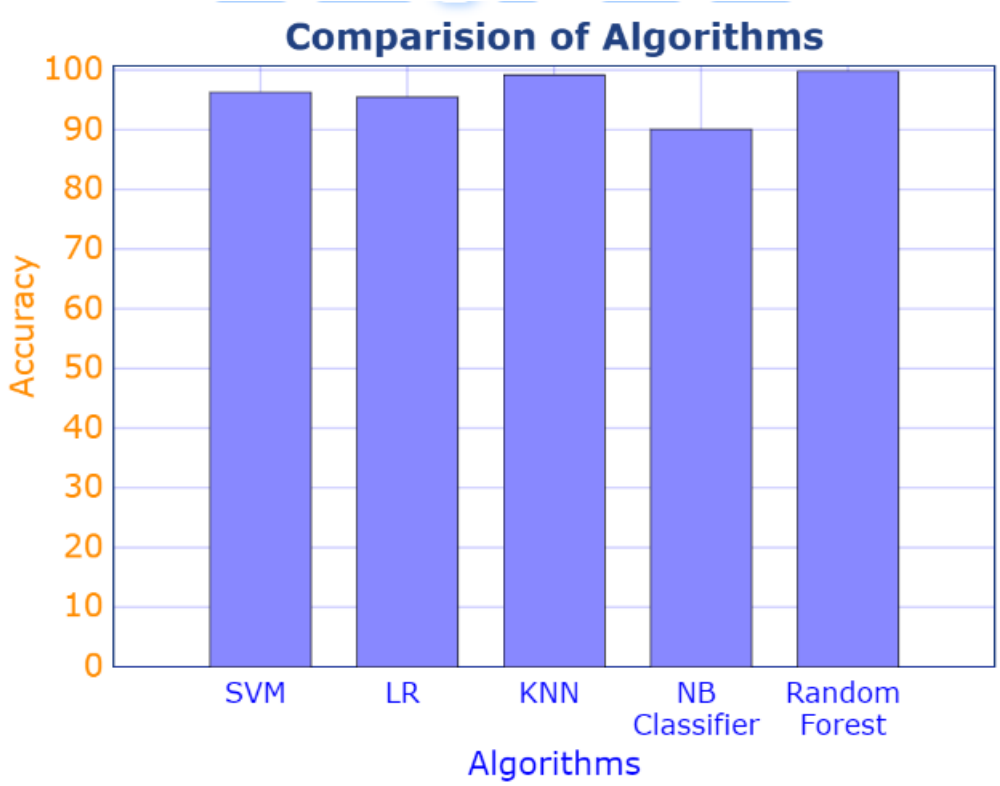


Figure.11 Confusion Matrix for Random Forest



8.Recommendations

- Software-based security systems are able to provide adequate protection against new forms of threats. After a period of time, the behaviour of the monitored environment may change, necessitating system retraining.

- If malicious behaviour is present in the training set, the system will treat it as anomaly. The precision will be higher if false alarm rates are low.

- We can use the deep learning and artificial neural networks for better accuracy of the machine learning models

9. Conclusion

The experiments showed that the classifiers can handlelarge amounts of data while stillproducing reliable results. For models that use the Random Forest Classifier, feature selectionandcleaningarecrucial.TheaccuracyofresultsusingRandomForestisaround0.997,whichissignificantlygreate

rthanthe0.9067accuracyofNaiveBayes.TheresultsshowthattheRandomForest classifier outperforms the Naive Bayes classifier in terms of ability and accuracy. IncomparisontoNaivebayes,therandomforesttakeslesstimetotrainandtestthedataset.Whencomparedto theother algorithms, theaccuracy of the resultsproducedbyNaiveBayesislower.Themajorgoalofthisstudywastodeterminetheconditionsunderwhichmodelsusingtheusing NSL-KDDdataset could attainclassification and accuracy.

Acknowledgments

The author wishes to express his gratitude to Dr.G.Suresh Reddy for his advice and ideas. He is the author's guide as well as the Head of the Information Technology Department at VNR VJIET in Hyderabad.

References (APA)

[1] Deepika Pandey Sanjay Pal (2018)"Intrusion Detection in Computer Networks By using Random Forest Algorithm Deepika Pandey Sanjay Pal" ISSN NO: 1076-5131

[2] M. Gupta (2015) Hybrid intrusion detection system: Technology and development,.International Journal of Computer Applications.

[3] A. P. Singh (2016) Analysis of host-based and network-based intrusion detection system.Abliz, M. (2011). Internet denial of service attacks and defense mechanisms. University of Pittsburgh ,Department of Computer Science.

[4] J. G. Noraini, M. R. (2011). Genetic algorithm performance with different selection strategiesin solving tsp. World Congress on Engineering. World Congress on Engineering.

[5] Abliz, M. (2011). Internet denial of service attacks and defense mechanisms. University of Pittsburgh,Department of Computer Science.

[6] S. Paliwal, R. G. (2012). Denial-of-service, probing & remote to user (r2l) attackdetection using genetic algorithm. International Journal of Computer Applications,, 57-62.

[7] V. Shmatikov, M.-H. W. (2007). ecurity against probe-response attacks in collaborativeintrusion detection. 2007 Workshop on Large Scale Attack Defense,ser. LSAD '07 (pp. 129-136). NY: ACM.

[8] T. M. I. White Paper. (2012). *Addressing big data security challenges: The righttools for smart protection*. Retrieved from Trend Micro: https://www.trendmicro.com/en_us/business.html[9] A. Elike Hodo, X. J. (2017). Shallow and deep networks intrusion detection system: A taxonomy and survey,. Retrieved from CoRR: https://arxiv.org/abs/1708.07174

[10] google developers. (2019). Machine learning crash course. Retrieved from google developers: https://developers.google.com/machine-learning/crash- course/classification/true-false-positive-negative

[11] Özgür A, E. H. (n.d.). . A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015. PeerJ Preprints. PeerJ Preprints.

[12] V. N. Vapnik, The nature of statistical learning theory. New-York: Springer Verlag, 1995.