Quest Journals
Journal of Software Engineering and Simulation
Volume 8 ~ Issue 9 (2022) pp: 23-33
ISSN(Online) :2321-3795 ISSN (Print):2321-3809
www.questjournals.org

Research Paper

Performance of Dimensionality Reduction and Machine Learning Model on Network Intrusion Datasets

EMMAH, Victor Thomas*
BENNETT, Emmanuel Okonni*
TAYLOR, Onate Egerton*
**Rivers State University, Port Harcourt, Nigeria*

Abstract-*With the increase in the utilization of computer network system, cyber-criminals have made computer network system their major target. This has made them carry out various forms of attacks such as Ddos, Dos, PortScan etc. Because of the constant penetration of cyber-attacks on a network system, many researchers have carried out various means of identifying and preventing these attacks by providing a robust model. Despite the provision of a robust system, cyber-criminals still have their way due to false classifications of network intrusion. This is a result of high dimensional data which give rise to the poor performance of machine learning models. For this problem to be solved, this paper identifies a machine learning design and a dimensionality reduction technique on three network intrusion datasets. The dimensionality technique used here is Principal Component Analysis (PCA), which was employed to select the most important features on the datasets. The reduced features were then used as our training dataset. The Random Forest Classifier (RFC) was applied to the training data using 100 nodes. After training, the performance of the model was then evaluated using a confusion matrix and a classification report. The evaluationresult shows that our RFC model achieved an accuracy result of 94%, and precision of 97% for the non-intrusion attack, 90% for Ddos attack, 87% for PortScan attack, and 90% for the Dos attack.*

Keywords- *Network Intrusion, Dimensionality Reduction, Random Forest Classifier, Network Security, Datasets*

R*eceived 04 Sep., 2022;* ***R****evised 17 Sep., 2022;* ***A****ccepted 19 Sep., 2022 © The author(s) 2022.*
P*ublished with open access at* ***www.questjurnals.org***

I. Introduction

The full big data explosion has definitely influenced believes that analyzing more data gives better results. It is true that machine learning models work better by learning more rules which helps in generalizing new data. Nevertheless, having low-quality input featuresand low-quality data give rise tonoisy data and also tends to make the training algorithm very slow. Consequently, when using large dataset with very high number of features for training, it is paramount to have an understanding of the maximum number of dataset features needed for effective and efficient training. Majority of these features have little or insignificant roles in the outcome of the training model. Various data-dimensionality reduction techniques are available to evaluate how informative each feature is and extract it from the dataset if necessary. (Silipo & Widmann, 2019).

In the early phase of machine learning, many features are added in order to obtain appropriate indicators and also to get a more precise result. Nonetheless, having an increased number of dataset features causes the training features of the dataset to increase enormously, thereby making analysis more difficult and also decreasing the model's output beyond a certain level. The above statedproblem is called "Curse of Dimensionality" (Hinton & Salakhutdinov, 2006). The presence of this problem is as a result of the enormous decrease in sample density as the dimensionality increases. Hence, if the dimensionality of the feature space is reduced, then this problem can be avoided. Dimensionality reduction is a technique of reducing with consideration the dataset features while obtaining a collection of the primary features. The selection attempts to choose a subset of the original features which can be used in the machine learning model. This allows obsolete and redundant features to be removed without much data loss. (Emmah *et al*., 2021).

Dimensionality reduction means the technique of reducing the number of attributes or features in a dataset while retaining as much variation as possible in the original dataset. This is done prior to training a model during the data preprocessing of the dataset. When the dataset dimensionality is reduced, a certain percentage of the dataset features are lost, based on the variance of the original data. Nonetheless, despite the loss of some data, dimensionality reduction has some merits. A small number of features inthe dataset reduces training time and computational resources and also enhances the general performance of machine learning algorithms. This is because models become more complex and tend to over-fit the training data when there are many features in the data, **so** reducing the dimensionality avoids the over-fitting problem.Also, reducing Dimensionality is a key factor in data visualization becausewhen the higher dimensional data is reduced to two or three components, then the data can be easily represented in a 3D or 2D plot. Reducing the Dimensionality of a large size of zero day dataset has some merits. When there are fewer attributes in the dataset, the time and space requirements for analyzing the data is reduced. When the data is free from multicollinearity, the parameters of the machine learning model is much easier to interpret and also visualization of the data during analysis becomes easy. This is because of the elimination of the noisy data which would have caused serious difficulty in interpreting the results (Singh, 2020).

By retaining the features which are of importance and discarding the redundant features, the noise in the data is removed. This enhances the accuracy of the model (Pramoditha, 2021).

II. Review of Related Works

Reddy, *et al*.,(2020) presented two dimensionality reduction approach after studying how they worked on Big Data.The Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA) were applied on four well-known machine learning algorithms;Decision Tree, Random Forest classifier,Support Vector Machine(SVM), and Naïve Bayes Classifier using the Cardiotocography (CTG) dataset. The result of the experiment showed that Principal Component Analysis outmatches Linear Discriminant Analysis in all measures used for the analysis as it gaveprecisions of 98.3%, 98.1%, 98.1% and 95% when applied with Random forest, SVM, Decision Tree and Naïve Bayes respectively with the dataset features reduced to 12; whereas the Linear Discriminant Analysis with dataset features reduced to 1 gave precisions of 97.4%, 97.8%, 85.6% and 97.4% for similar machine learning algorithms. Furthermore, similar dimensionality reduction methods were used in Intrusion Detection System (IDS) dataset and Diabetic Retinopathy (DR) dataset. The results derived indicated that the Principal Component Analysis gave the best precision, sensitivity and specificity on Random Forest, whereas the Support Vector Machine performed best when Linear Discriminant Analysis is employed on the Diabetic Retinopathy dataset.

Emmah, *et al.,* (2021) presented a comparative analysis of dimensionality reduction techniques on datasets for zero-day attack vulnerability analysis. Pareto-based Monte-Carlo filtering rule (PB-MCFR), Principal Component Analysis (PCA) and Truncated singular value decomposition technique (TruncatedSVD)were applied on malware dataset which was employed to analyze zero-day exploits. Afterdimensionality reduction has been performed on the attack dataset, the support vector classifier (SVM) was employed in training the model to identify zero-day vulnerability. The results derived from the analysis indicated that the PB-MCFR has the best prediction accuracy of 100% when compared with the TruncatedSVD and PCA which had an accuracy of 97% and 93% respectively. The results indicates that the Pareto Based-Monte Carlo approach has a better optimum performance than the TruncatedSVD and PCA with regards to dimensionality reduction for examining datasets for zero-day attack vulnerability.

Silipo and Widmann (2019) employed ten (10) approaches for dimensionality reduction. Thesetechniques were applied to the 2009 KDD Cup corpus dataset. The techniques used for dimensionality reduction on the Cup corpus datasetincludes:LDA,Baseline, Missing Values ratio, High Correlation filter,Low Variance filter, PCA,Random forest/Ensemble trees,forward feature construction + missing values ratio, Backward feature elimination+ missing values ration, Autoencoder and t-SNE. Some machine learning models were trained using the Cup corpus dataset and was then compared for performance based onaccuracy,reduction rate and area under the curve. With regards toreduction rate and overall accuracy, the random forest-based approach performed better with a precision of 90% which proves that it is most effective inretaining information for the classification task and removing uninteresting columns.

AlEroud & Karabatis (2013) employed the linear Data Transformation and anomaly detection methods to identify zero-day attacks. These methods were used on some common attack signatures which showcases contextual properties. The anomaly detection method uses the One-Class Nearest Neighbor (1-class NN) algorithm to identify zero-day attacks.To achieve dimensionality reduction, the One-Class Nearest Neighbor (1-class NN) algorithm was implemented using the singular value Decomposition (SVD) technique. This technique was evaluated using data gotten from the NSL-KDD intrusion detection dataset; and the outcome of the evaluation indicated that the singular value Decomposition played a major part in dimensionality reduction and subsequently provides for efficient and effective performance in identifying zero-day attacks.

Rathore *et al*., (2019) used different machine learning techniques and deep learning models foridentifying malware. To resolve the issue of curse of dimensionality in the feature vector, they made use of an optimal number of features using different variants of dimensionality reduction methods namely; Variance Threshold (VT) and Auto-Encoders with single layer (AE-1L) and 3-layer (AE-3L). The outcome of the various feature reduction methods indicated that Variance Threshold combined with Random Forest had the best precision of 99.78% which is slightly higher than no reduction (NONE and Random forest, 99.74%). AE-1L compared to deeper Auto-Encoder with 3 layers (AE-3L) had better performance with accuracy of 99.41% when combined with Random Forest. AE-3L based reduction had the lowest performance for all the methods. Variance Threshold (and Random Forest) achieved highest True Positive Rate (TPR) of 99.59%, while No feature reduction (None and Random forest) achieved highest True Negative Rate (TNR) of 100%.

Alkhayrat *et al.,* (2020) explored dimensionality reduction on a real telecom dataset and evaluate customers' clustering in reduced and latent space, compared to original space in order to achieve better quality clustering results. They used real-time data from SyriaTel Telecom Company that contains 220 features from 100,000 customers. The process of dimensionality reduction was performed to reduce its dimension to 20 features using both PCA decomposition and Autoencoder Neural Network built with Keras TensorFlow model to perform clustering analysis with unlabeled datasets. k-Means Clustering algorithm was then applied and evaluated on data in original and reduced space of features. The effects of dimensionality reduction was measured and compared with the two approaches, PCA and Autoencoder. The Clustering Performance was evaluated using Internal indices such as Silhouette index and Davies–Bouldin index. The best results for this type of data were obtained by autoencoder neural network approach which played a significant role in the dimensional reduction, and K-Mean Clustering Algorithm, where the clustering performance was enhanced with reduced dimensions.

III. Methodology

Figure 1 displays a sequential approach where three different network intrusion dataset were obtained and preprocessed. After preprocessing, dimensionality reduction was performed on the datasets so as to get a sparse data for efficient training result. The section below explains the different steps that are performed.

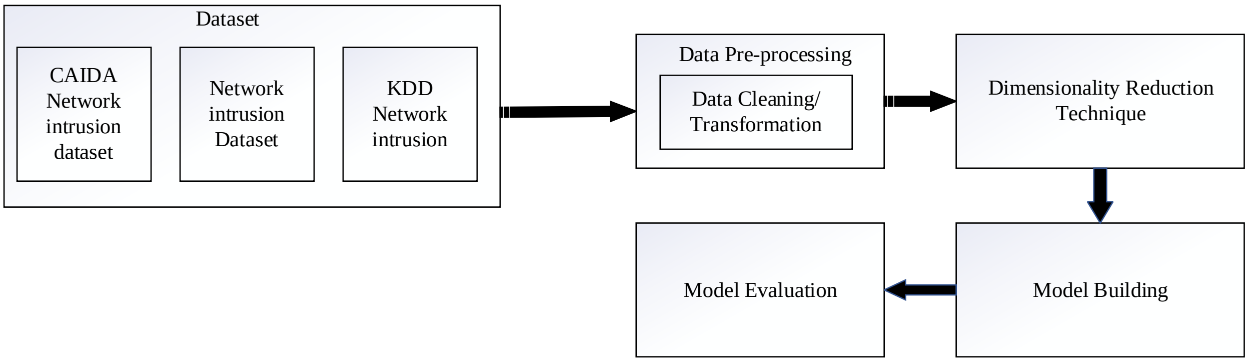
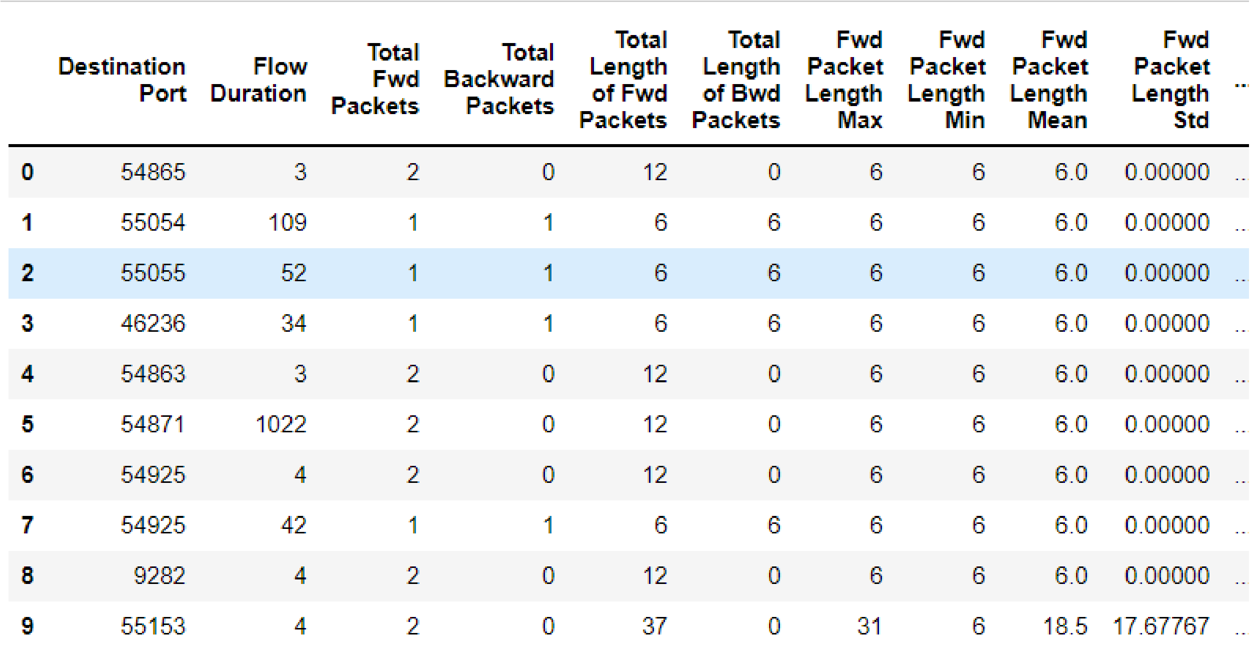

Dataset
CAIDA Network intrusion dataset
Network intrusion Dataset
KDD Network intrusion
Data Pre-processing
Data Cleaning/ Transformation
Dimensionality Reduction Technique
Model Building
Model Evaluation

Figure 1: Architectural Design

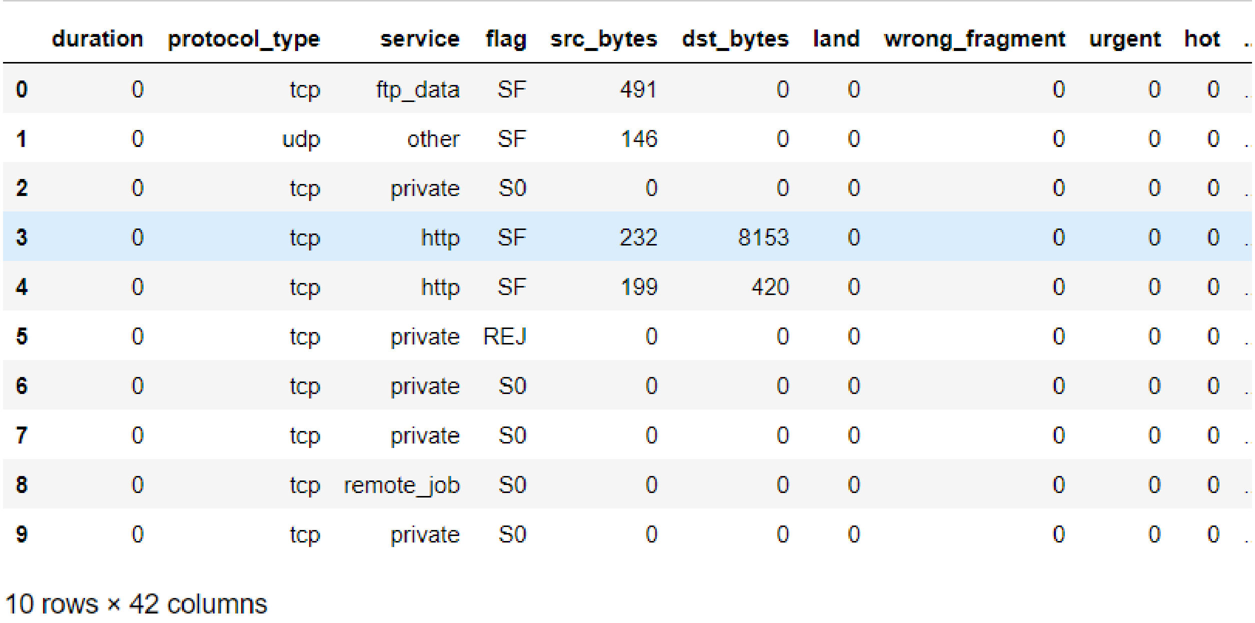
Dataset: Three network intrusion datasets were employed in this experiment and all downloaded from Kaggle.com. The first dataset comprises of 79 columns starting from the destination port column down to the class column. The class column in the dataset comprises of three different network intrusion attacks and a normal network flow. The three network intrusion attacks are DDoS, PortScan and Dos. The first 10 columns and 10 rows of the dataset can be seen in table 1.



	Destination Port	Flow Duration	Total Fwd Packets	Total Backward Packets	Total Length of Fwd Packets	Total Length of Bwd Packets	Fwd Packet Length Max	Fwd Packet Length Min	Fwd Packet Length Mean	Fwd Packet Length Std	..
0	54865	3	2	0	12	0	6	6	6.0	0.00000	..
1	55054	109	1	1	6	6	6	6	6.0	0.00000	..
2	55055	52	1	1	6	6	6	6	6.0	0.00000	..
3	46236	34	1	1	6	6	6	6	6.0	0.00000	..
4	54863	3	2	0	12	0	6	6	6.0	0.00000	..
5	54871	1022	2	0	12	0	6	6	6.0	0.00000	..
6	54925	4	2	0	12	0	6	6	6.0	0.00000	..
7	54925	42	1	1	6	6	6	6	6.0	0.00000	..
8	9282	4	2	0	12	0	6	6	6.0	0.00000	..
9	55153	4	2	0	37	0	31	6	18.5	17.67767	..

Table1 Network Intrusion dataset1

The second dataset comprises of 42 columns. The columns begins at duration, network protocol down to class column. The class column comprises of two categories, which are anomalous network flow and normal network flow. The first 10 columns and rows of the dataset can be seen in table 2.

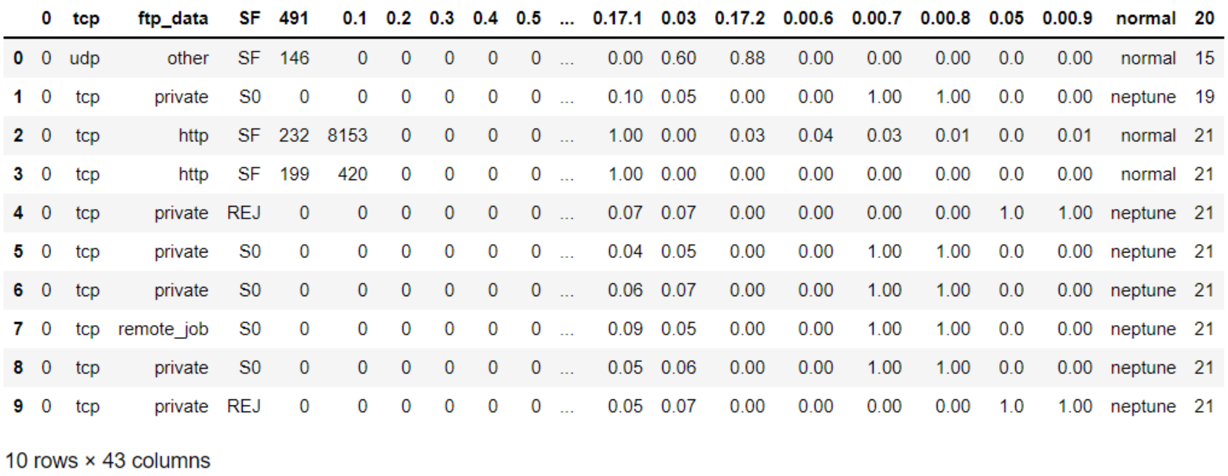


	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	.
0	0	tcp	ftp_data	SF	491	0	0	0	0	0	.
1	0	udp	other	SF	146	0	0	0	0	0	.
2	0	tcp	private	S0	0	0	0	0	0	0	.
3	0	tcp	http	SF	232	8153	0	0	0	0	.
4	0	tcp	http	SF	199	420	0	0	0	0	.
5	0	tcp	private	REJ	0	0	0	0	0	0	.
6	0	tcp	private	S0	0	0	0	0	0	0	.
7	0	tcp	private	S0	0	0	0	0	0	0	.
8	0	tcp	remote_job	S0	0	0	0	0	0	0	.
9	0	tcp	private	S0	0	0	0	0	0	0	.

10 rows × 42 columns

Table 2: Network Intrusion dataset 2

The third network intrusion dataset comprises of 43 columns starting from tcp column down to the normal column. The normal column is made up of 22 network intrusion attack and a normal network flow. A pictorial view of the dataset can be seen in table 3.



	0	tcp	ftp_data	SF	491	0.1	0.2	0.3	0.4	0.5	...	0.17.1	0.03	0.17.2	0.00.6	0.00.7	0.00.8	0.05	0.00.9	normal	20
0	0	udp	other	SF	146	0	0	0	0	0	...	0.00	0.60	0.88	0.00	0.00	0.00	0.0	0.00	normal	15
1	0	tcp	private	S0	0	0	0	0	0	0	...	0.10	0.05	0.00	0.00	1.00	1.00	0.0	0.00	neptune	19
2	0	tcp	http	SF	232	8153	0	0	0	0	...	1.00	0.00	0.03	0.04	0.03	0.01	0.0	0.01	normal	21
3	0	tcp	http	SF	199	420	0	0	0	0	...	1.00	0.00	0.00	0.00	0.00	0.00	0.0	0.00	normal	21
4	0	tcp	private	REJ	0	0	0	0	0	0	...	0.07	0.07	0.00	0.00	0.00	0.00	1.0	1.00	neptune	21
5	0	tcp	private	S0	0	0	0	0	0	0	...	0.04	0.05	0.00	0.00	1.00	1.00	0.0	0.00	neptune	21
6	0	tcp	private	S0	0	0	0	0	0	0	...	0.06	0.07	0.00	0.00	1.00	1.00	0.0	0.00	neptune	21
7	0	tcp	remote_job	S0	0	0	0	0	0	0	...	0.09	0.05	0.00	0.00	1.00	1.00	0.0	0.00	neptune	21
8	0	tcp	private	S0	0	0	0	0	0	0	...	0.05	0.06	0.00	0.00	1.00	1.00	0.0	0.00	neptune	21
9	0	tcp	private	REJ	0	0	0	0	0	0	...	0.05	0.07	0.00	0.00	0.00	0.00	1.0	1.00	neptune	21

10 rows × 43 columns

Table 3: Network Intrusion dataset 3.

Data-Preprocessing: By data pre-processing we mean converting some columns in the dataset into a language or format that the machine learning model can understand the dataset perfectly for achieving a better accuracy. Pre-processing can also be seen as the process of removing missing values and duplicate values. We used pandas library available in python programming language in reading the various intrusion dataset into our working directory. We also performed data cleaning on the various intrusion datasets by using dataset.isnull(), to check and remove null values and removal of non-alphanumeric characters.

Dimensionality Reduction Technique: Due to the problems of high dimensional data, Principal Component Analysis (PCA) approach was utilized in lowering the complexities in a fuzzy system because the dataset capabilities expand from factor i_1 ... i_n, wherein i= input characteristic and n=1,2,3 … 8. Principal Component Analysis (PCA) is a matrix factorization approach that generalizes the Eigen decomposition of a rectangular matrix (m x n) to any matrix (n x m). Principal Component Analysis(PCA) primarily based totally to reduce high dimensional data.

Algorithm for Principal Component Analysis

Step 1: Obtain the dataset

Step 2: Calculate the covariance matrix for the functions in the dataset.

Step 3: Calculate the eigenvalues and eigenvectors for the covariance matrix.

Step 4: Sort eigenvalues and their corresponding eigenvectors.

Step 5: Pick okay eigenvalues and shape a matrix of eigenvectors.

Step 6: Transform the unique matrix.

Pseudocode for PCA

```
system PCA
Compute dot product matrix: XTX
Eigenanalysis: XTX = V?VT
Compute eigenvectors: U = XV?
Keep particular range of first components: Ud = [u1,...,ud]
```

6: Compute d functions: Y = UdTX

Model Development: The model was developed using Random Forest Classifier. The model was trained by passing 70% of each of the network intrusion dataset to the random forest classifier and 30% of the dataset will be used for testing. In other to get a better training accuracy, we used the number of estimators to be 100. The number of estimators denotes the connections between the nodes in the random forest classifier.

Model Evaluation: Confusion matrix and classification report was used in evaluating the performance of the model. The model's performance was evaluated based on accuracy, precision, recall, total number of correct predictions and total number of incorrect predictions.

IV. Experiments

An experiment was conducted by performing a simulation in Jupyter Notebook technology. The jupyter notebook allows users to enter various parameters and perform simulation analysis. In the jypyter notebook, we presented a comparative analysis of various intrusion datasets using a machine learning algorithm and a dimensionality reduction technique. The machine learning algorithm used here is Random Forest Classifier (RFC), and Principal Component Analysis (PCA) was used as a dimensionality reduction technique. We used three network intrusion datasets. The datasets comprise of over 40 features. The datasets also comprise of different types of attacks that is being carried out on computer networks. Some of these attacks are DDoS, PortScan, and Dos. A count plot that shows the different types of intrusion attacks and the number of times they occurred in the dataset can be seen in figure 2, 3, and 4. Before we proceed to building our RFC model in detecting network intrusion attacks, we first perform some data cleaning and pre-processing, which has to do with the removal of noise, and filling or removal of null values. We also convert all alphabets to numbers using the LabelEncoder function. After these processes, we applied PCA each of the datasets in selecting the most important features, as well as reducing the features of the datasets without distorting the content of the data. The reduced features was then used as our training dataset. The RFC was applied on the training data using 100 of nodes. After training of the RFC, we evaluated the performance of our model using a confusion matrix and a classification report. The confusion matrix is used to show the number of correct prediction vs the predicted result while the classification report is used to shows the accuracy level, precision, f1-measure and recall.

V. Results and Discussion

Figure 5shows the classification report of the first intrusion dataset. The result shows that our RFC model achieved an accuracy result of 94%, and precision of 97% for non-intrusion attack, 90% for Ddos attack, 87% for PortScan attack and 90% for Dos attack. Figure6 shows the confusion matrix of the first dataset. The result of the confusion matrix shows the number of correct predictions. This was used in showing how many times the RFC model predicted the classes correctly. Figure 7 and 8 shows the classification report of the second intrusion dataset. The result of the classification report shows that our RFC model had an accuracy result of 99%, and a precision level of 99% for non-attacks and 99% for network attacks. Figures9 and 10 show the classification report and confusion matrix of the third dataset. The result of the confusion matrix shows that our RFC model achieved an accuracy result of 98% and a precision level of 98% for non-attacks and 99% on network attacks. In general, the results of the RFC model on all the three network intrusion dataset shows that RFC model with PCA will be more good and suitable in building a robust system for detecting and preventing network intrusions.

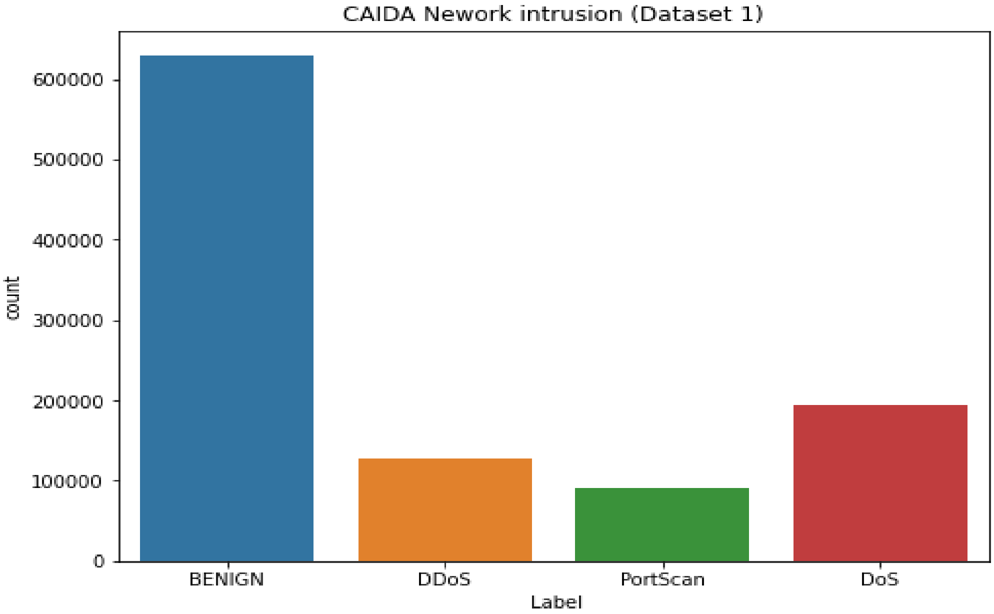


Figure 2: Countplot that depicts the number of occurrences of non-attacks and attacks that was carried out on a network system. Figure 2denotes the categories of attacks that can be seen in an intrusion system. that of the first dataset

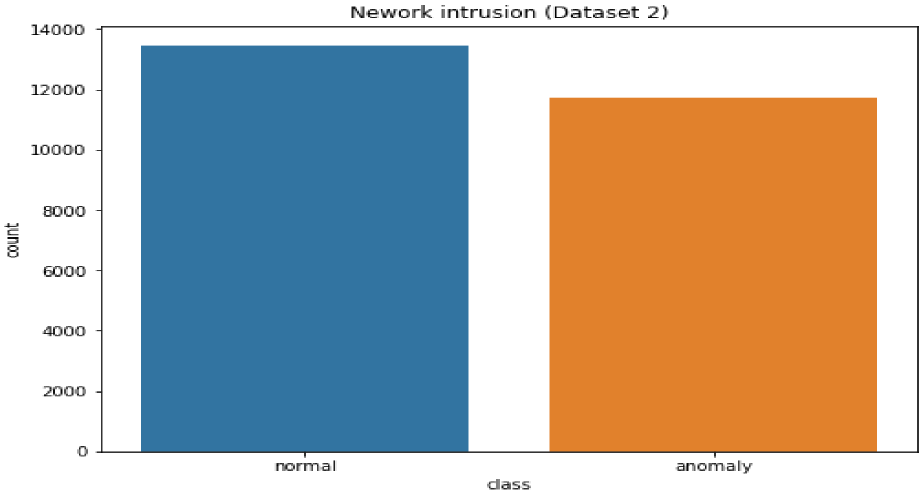


Figure 3: Countplot that depicts the number of occurrence of non-attacks and attacks that was carried out on a network system. This is for the second dataset

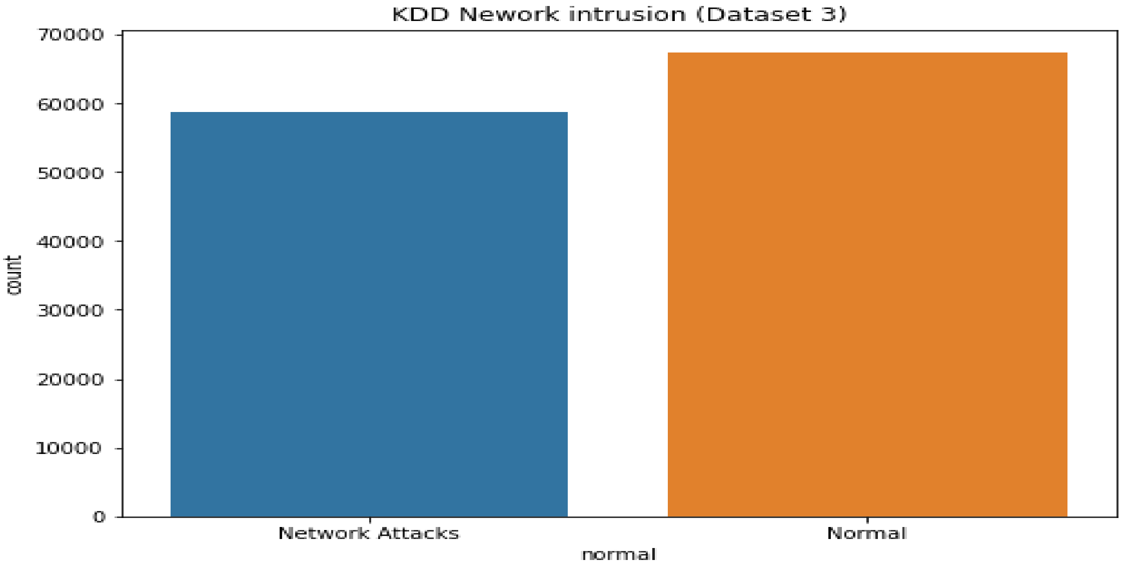
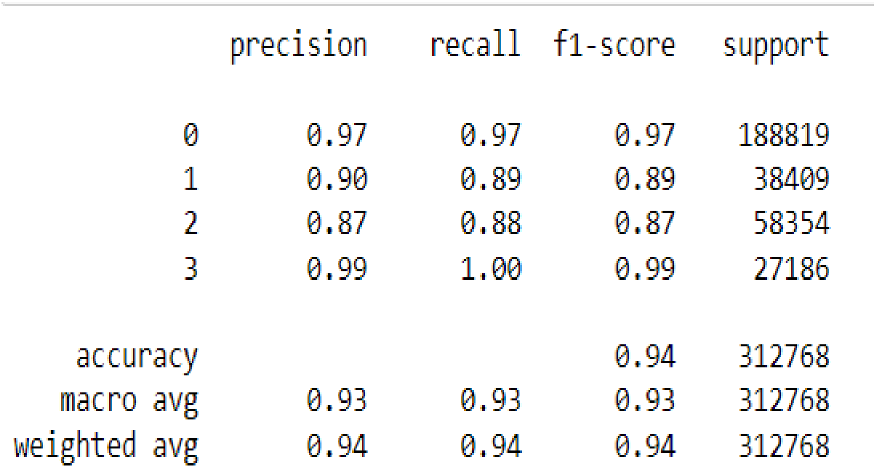


Figure 4: Countplot that depicts the number of occurrence of non-attacks and attacks that was carried out on a network system.

Figure 4denotes that of the third dataset (KDD) dataset.



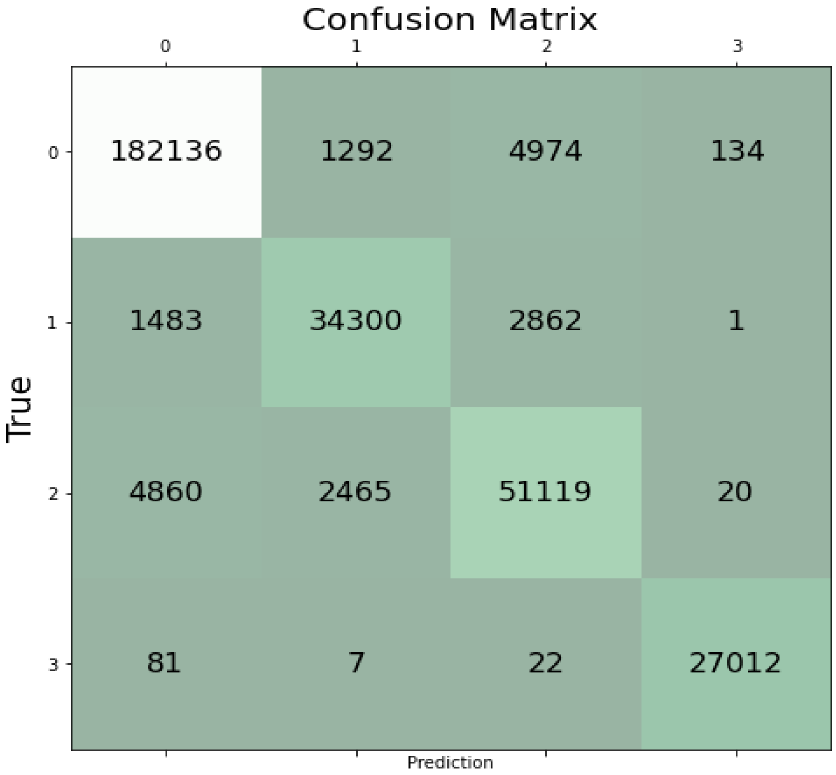
```
              precision    recall  f1-score   support

           0       0.97      0.97      0.97    188819
           1       0.90      0.89      0.89     38409
           2       0.87      0.88      0.87     58354
           3       0.99      1.00      0.99     27186

    accuracy                           0.94    312768
   macro avg       0.93      0.93      0.93    312768
weighted avg       0.94      0.94      0.94    312768
```

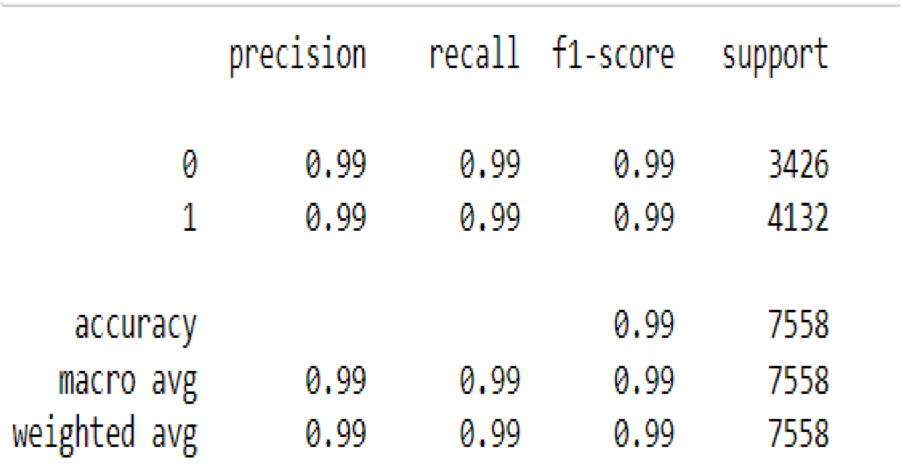
Figure 5: Classification report of our RFC model on the first dataset

The classification report depicts the accuracies and precisions of the four different categories. The result of the classification report shows that the accuracy for normal network packets are 97%, DDos (90%), Portscan (87%), and Dos (99%).



True \ Prediction	0	1	2	3
0	182136	1292	4974	134
1	1483	34300	2862	1
2	4860	2465	51119	20
3	81	7	22	27012

Figure 6: Confusion Matrix of our RFC model on the first dataset
This shows the total number of correct predictions and the total number of incorrect preedictions.



```
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      3426
           1       0.99      0.99      0.99      4132

    accuracy                           0.99      7558
   macro avg       0.99      0.99      0.99      7558
weighted avg       0.99      0.99      0.99      7558
```

Figure 7: Classification report of our RFC model on the second dataset

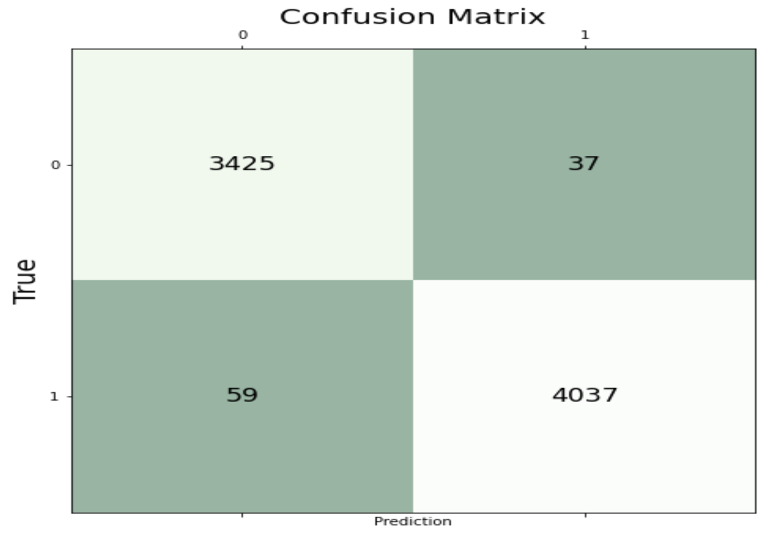
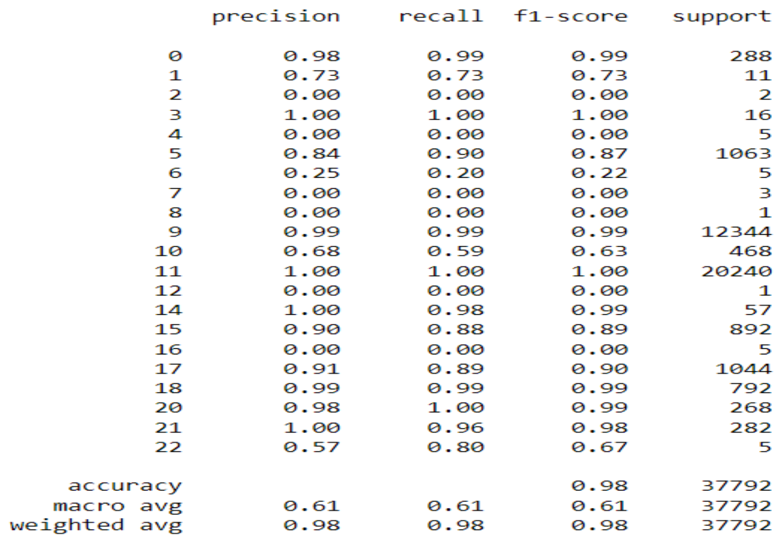


Figure 8: Confusion Matrix of our RFC model on the second dataset



```
              precision    recall  f1-score   support

           0       0.98      0.99      0.99       288
           1       0.73      0.73      0.73        11
           2       0.00      0.00      0.00         2
           3       1.00      1.00      1.00        16
           4       0.00      0.00      0.00         5
           5       0.84      0.90      0.87      1063
           6       0.25      0.20      0.22         5
           7       0.00      0.00      0.00         3
           8       0.00      0.00      0.00         1
           9       0.99      0.99      0.99     12344
          10       0.68      0.59      0.63       468
          11       1.00      1.00      1.00     20240
          12       0.00      0.00      0.00         1
          14       1.00      0.98      0.99        57
          15       0.90      0.88      0.89       892
          16       0.00      0.00      0.00         5
          17       0.91      0.89      0.90      1044
          18       0.99      0.99      0.99       792
          20       0.98      1.00      0.99       268
          21       1.00      0.96      0.98       282
          22       0.57      0.80      0.67         5

    accuracy                           0.98     37792
   macro avg       0.61      0.61      0.61     37792
weighted avg       0.98      0.98      0.98     37792
```

Figure 9: Classification report of our RFC model on the third dataset

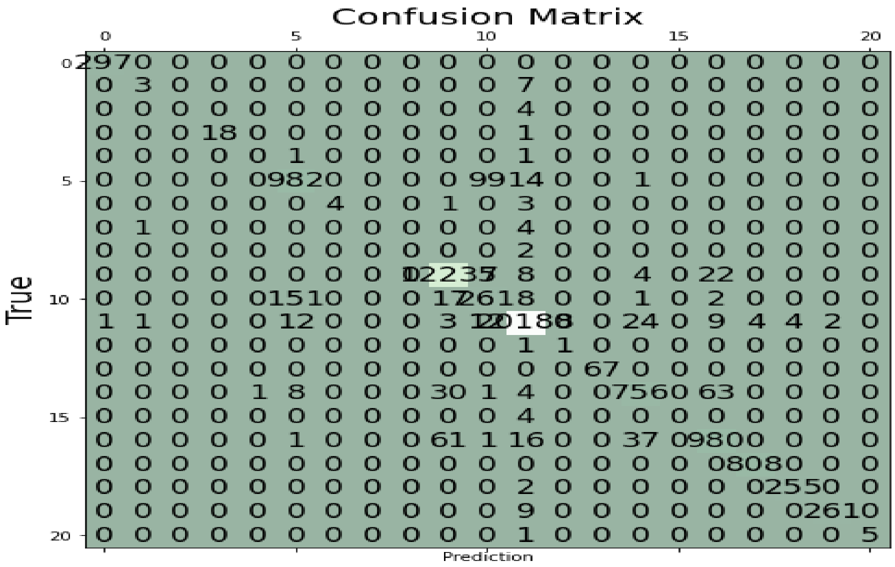


Figure 10: Confusion Matrix of our RFC model on the third dataset

VI. Conclusion and Future Work

With the increase in the utilization of computer network system, cyber-criminals has made computer network system their major target. This has made them carry out various forms of attacks such as DDoS, DoS, PortScan etc. Because of the constant penetration of cyber-attacks on a network system, many researchers have carried out various means of identifying and preventing these attacks by providing a robust model. Despite the provision of a robust system, cyber-criminals still have their way due to false classifications of network intrusion. This is a result of high dimensional data which gave rise to the poor performance of machine learning models. In order to solve this problem, applied a dimensionality reduction technique on three network intrusion datasets. The dimensionality reduction technique was used in selecting the most important features of the network intrusions dataset. We then used the reduced features in training a random forest model. We also compared each of the results obtained from the three network intrusion dataset, in other to know which network intrusion dataset is more suitable and reliable for building a robust system for detecting and preventing intrusions on computer networks. This work can be improved by utilizing more than one machine learning algorithm on the network intrusions dataset. This will give a more detailed explanation of which machine learning model is best suitable for network intrusions.

References

[1]. Silipo, R. and Widmann, M. (2019); 3 New Techniques for Data-Dimensionality Reduction in Machine Learning. Retrieved 21st march, 2022. https://thenewstack.io/3-new-techniques-for-data-dimensionality-reduction-in-machine-learning/

[2]. Pramoditha, R. (2021). 11 Dimensionality reduction techniques you should know in 2021. towards data science.

[3]. Singh, P. (2020). Dimensionality Reduction Approaches: Ways of obtaining principle variable for better data representation, improving efficiency, and saving time. https://towardsdatascience.com/dimensionality-reduction-approaches-8547c4c44334

[4]. Emmah, V. T., Ugwu, C., & Onyejegbu, L. (2021); Comparative Analysis of Dimensionality Reduction Techniques on Datasets for Zero-Day Attack Vulnerability.

[5]. Reddy, G. T., Reddy, M. P. K., Lakshmanna, K., Kaluri, R., Rajput, D. S., Srivastava, G., & Baker, T. (2020). Analysis of dimensionality reduction techniques on big data. IEEE Access, 8, 54776-54788.

[6]. Aleroud, A., & Karabatis, G. (2013). Toward zero-day attack identification using linear data transformation techniques. In 2013 IEEE 7th International Conference on Software Security and Reliability (pp. 159-168). IEEE.

[7]. Rathore, H., Agarwal, S., Sahay, S. K., & Sewak, M. (2018, December). Malware detection using machine learning and deep learning. In International Conference on Big Data Analytics (pp. 402-411). Springer, Cham.

[8]. Alkhayrat, M., Aljnidi, M., & Aljoumaa, K. (2020). A comparative dimensionality reduction study in telecom customer segmentation using deep learning and PCA. Journal of Big Data, 7(1), 1-23.

[9]. Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. Journal of Science, 313(5786), 504-507.