Quest Journals
Journal of Software Engineering and Simulation
Volume 7 ~ Issue 8 (2021) pp: 48-56
ISSN(Online) :2321-3795 ISSN (Print):2321-3809
www.questjournals.org

Research Paper

Comparative Analysis of Dimensionality Reduction Techniques on Datasets for Zero-Day Attack Vulnerability

Victor Emmah Thomas[1], Chidiebere Ugwu[2], Laticia Onyejegbu[2]
[1]*Department of Computer Science, Rivers State University*
[2]*Department of Computer Science, University of Port Harcourt*
Rivers State, Nigeria

ABSTRACT— *This paper presents a comparative analysis of three dimensionality reduction techniques on a dataset for zero-day vulnerability analysis. The comparison was carried out in terms of classification report (which shows the percentage of accuracy, precision and recall), and confusion matrix (which shows the true predicted level vs the original label data). The dimensionality reduction techniques compared in this paper are principal component analysis (PCA), truncated singular value decomposition technique (TruncatedSVD) and pareto-based Monte carlo technique (PB-MCFR). After the reduction in dimensionality of the attack dataset, a machine learning algorithm namely, support vector classifier (SVM) was used to train the model to detect zero-day vulnerability. The results obtained from the comparative analysis showed that the PB-MCFR achieved the highest accuracy in prediction of 100% as compared to the PCA and TruncatedSVD which gave 93% and 97% accuracies respectively. This proves that Pareto Based-Monte Carlo technique is better than the PCA and TruncatedSVD for dimensionality reduction in analyzing datasets for zero-day attack vulnerability.*
KEYWORDS: *Zero-Day vulnerability, Malware, Dimensionality Reduction, Sparse filtering, Malware analysis.*

Received 02 August, 2021; Revised: 14 August, 2021; Accepted 16 August, 2021 © The author(s) 2021. Published with open access at www.questjournals.org

I. INTRODUCTION

The internet today has become a persistent threat environment for various types of organisations. Every day as new technologies and sophisticated applications are developed, and are also being adopted to meet the changing needs of businesses, malicious sources lie in wait in other to exploit the vulnerabilities found in them. Zero-day and its associated attacks have dominated headlines over the years for political and socio-economic benefits. Malware have consistently been used to facilitate criminal activities, espionage within countries and industries, and other unwanted activities on our computer networks; hence, attackers have found malware to be a very important and key tool for their malicious campaigns. The time that exists between when a vulnerability in a computer system or software product is first exploited and taken advantage of, and when software developers and security experts starts to develop a response and thwart to that threat is known as the window of vulnerability [4].

For any type of zero day attack prediction and detection, datasets need to be analysed and classified. Zero day data involves large amount of malware datasets to be trained using machine learning algorithms in order to learn more rules and perform better generalization to new data. However, indiscriminate introduction of low quality data which contain redundant input features which may not be important for the training may introduce too much of noisy data and at the same time slow down the training considerably. Therefore, it is good practice to look at how many of these data features are really useful for the model. In machine learning, we tend to add as many features as possible at first, in order to get useful indicators and obtain a more accurate result. Nevertheless, as the number of dataset instances increase, the features for the training increase exponentially, thereby making the analysis to be difficult and decreasing the model's output after a certain level. This is known as the "Curse of Dimensionality". This problem exist because the density of the sample decreases exponentially as the dimensionality increases. We can therefore overcome this problem by reducing the dimensionality of the feature space. Reduction of dimensionality is the method of reducing with consideration the dataset features while obtaining a collection of the principal features. The selection tries to pick a subset of the original features to be used in the machine learning model. By doing this, redundant and obsolete features can be deleted without incurring much information loss.

Dimensionality reduction of large size of zero day dataset has some advantages. The time and space required to analyse the data is reduced when the dataset has fewer attributes. When the multicollinearity of the data is removed, it becomes easier to interpret the parameters of the machine learning model and also visualize the data using 2D or 3D during analysis. This is because the noisy data is removed which would have caused the difficulty in interpretation of the results [15].

The various approaches used to detect zero-day malware are divided into Statistical-based, Signature-based and Behaviour-based. **Signature-based** technique maintains a database of well-known malware. **Statistical-based** technique depends on attack profiles built from historical data **Behaviour-based** technique detect and classify malware based on their behaviour. Signature-based is easy to evade, statistical-based cannot profile new and emerging malware and behaviour based is slow [8]. However detecting these malware involves analysing a large dataset of malware samples of both known and unknown including their full features, some of which may not be necessary in the analysis. This leads to a problem encountered in big data analysis known as 'curse of dimensionality. This is a situation that arises when analysing and organising data in high dimensional space. The concern here is that as the number of malware samples increases, the number of possible distinct configurations also increases exponentially. In machine learning classification problems, there are often too many factors on the basis of which the final classification is done. These factors are variables that hinges on the data features. The higher the number of features, the harder it becomes to visualize the training set for analysis. Sometimes, most of these features are correlated, and hence redundant. This is the reason for the removal of redundant and null values which may not be necessary for the analysis in order to maintain a less noisy data. Dimensionality reduction is the process of reducing the number of random variables under consideration, by obtaining a set of principal variables. It can be divided into feature selection and feature extraction. Dimensionality is important as it helps in data compression, and hence storage, and also reduces the computational time.

II. REVIEW OF RELATED WORKS

There have been different literatures that studied malware analysis and also how to reduce the dimensionality of datasets used in zero-day attack analysis. Some of these related works are discussed in this section.

[13] designed a scalable approach towards discovery of unknown vulnerabilities. They designed a hybrid architecture framework for zero-day attack detection and risk level assessment with respect to likelihood of exploits. It consists of three phases namely, zero-day attack path analyser, Risk analyser and physical layer. The zero-day attack path analyser is liable to detect the unknown vulnerability, the risk analyser is assigned to analyse the generated attack and the physical layer consists of database and a centralized server that are used during the information processing of the first two layers. The framework follows a probabilistic approach for identification of the zero day attack path and further to rank the severity of the identified zero-day vulnerability. Experiments were performed using polymorphic engines namely, ADMutate, clet, Alpha2, CountDown, JumpCallAdditive and Pex to confirm the accuracy of the framework and the system was found to have 89% detection rate and 3% False Positive Rate (FPR) of zero-day attacks.

[8] proposed a hybrid real-time zero-day attack detection and analysis system which combines anomaly-based detection, behavior-based detection and signature-based detection techniques. The architecture proposed by [8] has three layers namely; Detection layer, Analysis layer and Resource layer. The system employs 1-class SVM as an anomaly detection technique in detection layer to detect zero-day attacks that diverts from the good traffic profile. The analysis layer in the system captures both static and dynamic behavior of malicious binaries captured in the detection layer. The analysis stub combines both static and dynamic malware analysis functionalities to work as a single unit in a component based architecture where any feature can be replaced in the future. The Static Analysis Engine (SAE) provides basic information to profile the malicious binary and Dynamic Analysis Engine (DAE) captures run-time behavior of a malicious binary by executing it in an emulator.

[2] presented machine learning methods for malware detection and classification. The purpose was to determine the best feature extraction, feature representation, and classification methods that result in the best accuracy when used on the top of Cuckoo Sandbox. Different classifiers were evaluated with 1156 malware file of 9 families of different types and 984 benign files of various formats. From the author's result, Random Forest method was recommended to implement the classification for Monte-class classification, as it resulted in the best accuracy and high performance.

[3] combined supervised and unsupervised learning for Zero-day Malware detection. In their work, they presented a novel machine learning based framework to detect known and newly emerging malware at a high precision using layer 3 and layer 4 network traffic features. Their framework leverages the accuracy of supervised classification in detecting known classes with the adaptability of unsupervised learning in detecting new classes. They proposed an architecture which consist of six major components namely, (i) Data capture, (ii)

An intrusion detection/prevention system, (iii) Information storage, (iv) feature extraction and transformation, (v) Supervised classifier, and (vi) a UI portal.

[10] developed SweetBait. It is a distributed system that is a combination of network intrusion detection and prevention techniques. It employs different types of honeypot sensors, both high interaction and low-interaction to recognize and capture suspicious traffic. SweetBait automatically generates signatures for random IP address space scanning worms without any prior knowledge. And for the non scanning worms, Argos is used to do the job. A novel aspect of this signature generation approach is that a forensics shellcode is inserted, replacing malevolent shellcode, to gather useful information about the attack process.

[1] proposed a contextual misuse and anomaly detection prototype to detect zero-day attacks. The contextual misuse detection utilizes similarity with attack context profiles, and the anomaly detection technique identifies new types of attacks using the One Class Nearest Neighbor (1-NN) algorithm. It uses information entropy and linear data transformation to generate feature-based and linear function-based attack profiles and systematically creates contextual relationships between known attacks to generate attack profiles that capture activities of zero-day attacks.

[11] discussed the Analysis of Dimensionality Reduction Techniques on Big Data by investigating two prominent dimensionality reduction techniques, namely, Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA) on four popular machine learning algorithms, Decision Tree Induction, Support Vector Machine((SVM), Naïve Bayes Classifier and Random Forest classifier using the Cardiotocography (CTG) dataset. Results from their experiments showed that PCA outperforms LDA in all the measures used for the analysis as it gave accuracies of 98.1%, 95%, 98.3% and 98.1% when used with Decision Tree, Naïve Bayes, Random forest and SVM respectively with reduced dataset features to 12; whereas LDA with reduced features to 1 gave accuracies of 97.4%, 85.6%, 97.4% and 97.8% for the same machine learning algorithms. Also, the same dimensionality reduction techniques were applied to Diabetic Retinopathy (DR) dataset and Intrusion Detection System (IDS) dataset. The results showed that PCA yielded best accuracy, specificity and sensitivity on Random Forest, whereas SVM performed best when LDA is applied on the DR dataset.

From most of the literatures reviewed, it is evident that a lot of the malware analysis have been carried out by using large data sizes (malware samples), including all the malware features. Our approach will show that some of the features of malware used for analysis are simply irrelevant and may not be necessary for the analysis and detection of unknown vulnerabilities. Hence, the introduction of sparse filtering approach for reduction of the size and features of the malware samples to easily analyze fewer malware to reduce the error in detecting the vulnerability and identify the attack.

III. METHODOLOGY

The methodology describes the processes on how the comparative analysis of three dimensionality reduction techniques on a Zero day malware dataset is carried out. The KDD dataset gotten from http://www.kaggle.com was used for this analysis. It is one of the most widely used dataset for analyzing intrusion detection systems (IDS). Each instance of the dataset contains 43 features, ranging from 'tcp' to 'normal' columns. The normal columns consist of different types of attack profiles ranging from Neptune attack to normal. The normal columns consist of 22 types of attacks in total. A sample of the dataset is shown in figure 1

	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 rows × 43 columns

Figure 1 Sample of KDD data for zero day attack

Principal Component Analysis (PCA): This is a statistical procedure that orthogonally transforms the original n numeric dimensions of a dataset into a new set of n dimensions. PCA is a linear dimensionality reduction technique (algorithm) that transforms a set of correlated variables (p) into a smaller k (k<p) number of uncorrelated variables called ***principal components*** while retaining as much of the variation in the original dataset as possible. In the context of Machine Learning (ML), PCA is an unsupervised machine learning algorithm which is used for dimensionality reduction. Algorithm 1 shows the steps to compute the PCA

Algorithm 1 Principal Component Analysis

1: **procedure** PCA
2: Compute dot product matrix: $\mathbf{X}^T\mathbf{X} = \sum_{i=1}^{N}(\mathbf{x}_i - \boldsymbol{\mu})^T(\mathbf{x}_i - \boldsymbol{\mu})$
3: Eigenanalysis: $\mathbf{X}^T\mathbf{X} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T$
4: Compute eigenvectors: $\mathbf{U} = \mathbf{X}\mathbf{V}\boldsymbol{\Lambda}^{-\frac{1}{2}}$
5: Keep specific number of first components: $\mathbf{U_d} = [\mathbf{u}_1, \ldots, \mathbf{u}_d]$
6: Compute d features: $\mathbf{Y} = \mathbf{U_d}^T\mathbf{X}$

Truncated Singular Value Decomposition (TruncatedSVD) is a dimension reduction technique for matrices that reduces the matrix into its component to simplify the calculation. It works well with sparse data in which many of the row values are zero. In contrast, PCA works well with dense data. Truncated SVD can also be used with dense data. Another key difference between truncated SVD and PCA is that factorization for SVD is done on the data matrix while factorization for PCA is done on the covariance matrix.

Algorithm 2 TruncatedSVD

```
Step1: Define a matrix A = array[]
Step2: Set U,s, VT = svd(A)
Step 3 Create a m x n Sigma matrix
        Sigma = Zeros((A.shape[0], A.shape[1]))
Step4: Populate the Sigma with n x n diagnonal mattrix
        Sigma[:A.shape[0], :A.shape[0]] = diag(s)
Setp5: Set:
     n_elements = 2
     Sigma = Sigma[:, :n_elements]
     VT = VT[:n_elements, :]
Step6: Reconstruct the matrix
      B = U.dot(Sigma.dot(VT))
Step7: Transform the Matrix
      T = U.dot(Sigma)
      print(T)
      T = A.dot(VT.T)
      print(T)
```

Pareto-Based Monte Carlo Filtering (PB-MCFR) is a mathematical procedure, which is used to filter 20% of the dataset features by applying the pareto rule and perfoming that for a finite number of trial runs called the monte carlo observation space. The idea is based on the 20/80 rule which states that 20% of the dataset features can yield 80% of the result expected. It is used to estimate the possible outcomes of an uncertain event of a dataset, and it can also be used for reducing the dimension of a given data.

Algorithm 3: PB-MCFR

```
Step1: Read Input
        for until SampleSize do
        Extract at random, 20% of network traffic data features.
                end for
Step2: Filter previous known vulnerability
for all NTDsparse = 20%(NTD)
        if (NTD_sparse == isbad)
        block: (NTD_sparse)
        else
        pass((NTD_sparse )
        endif
end for
isbad Logic:
if NTD_sparse == unknown &&
        isbad = True
else
```

isbad = False

end

Step3: Repeat Steps 1-2 till a desired number of searches have been made

---

IV. EXPERIMENTS AND RESULTS

In the experiment, the three dimensionality reduction techniques is applied seperately on the KDD dataset obtained to see which technique is more effective in sparse filtering of the dataset features. These dataset with sparse features obtained using each of the techniques is applied separately to a machine learning algorithm to train a model in other to detect a zero-day attack. The dataset contain values which have to be standardized and pre-processed in other to obtain a better training data. Pre-processing of the data is achieved by checking for missing values and duplicate values. Columns that have alphabets as values are converted to numeric digits starting from 0-22. This is achieved using the LabelEncoder function. We then split the dataset into a training and a testing data, where 70% of the data was used for training and 30% of the data was used for testing. The dimensionality reduction techniques used are Pareto-Based Monte-Carlo technique (PB-MCFR), Principal Component Analysis (PCA) and truncated singular value decomposition technique (TruncatedSVD). After these processes, we then applied the reduced features with the transformed data of each of the three dimensionality reduction technique to a Support Vector Classifier in training the model that will detect a zero day attack. After the training, the model performance on each of the reduced dimensionality techniques was noted based on metrics such as accuracy, precision false negative and false positive result. Figures 2 and 3 show classification report and the confusion matrix respectively of using the principal component analysis (PCA) for reduction in the dataset features and subsequent training of the model using the SVM. Figures 4 and 5 shows similar classification report and confusion matrix respectively when the TruncatedSVD is applied on the data and figures 6 and 7 show the results of classification report and confusion matrix when the PB-MCFR is used. In table 1, a summary of the performance metrices is presented for the three techniques, highlighting the precision, recall, f1-score and accuracy. The PB-MCFR is seen to produce the highest accuracy result of about 100%, followed by principal component analysis and truncated singular value decomposition technique, which both have an accuracy result of about 93% and 97% approximately. A graph that represents the model performance of the techniques is shown in figure 8.

	precision	recall	f1-score	support
0	0.87	0.09	0.16	763
1	0.00	0.00	0.00	26
2	0.50	0.29	0.36	7
3	0.67	0.05	0.09	41
4	0.62	1.00	0.76	8
5	0.67	0.86	0.75	2874
6	0.00	0.00	0.00	13
7	0.00	0.00	0.00	8
8	0.00	0.00	0.00	6
9	0.94	0.99	0.97	32971
10	1.00	0.17	0.29	1216
11	0.96	0.97	0.97	53807
12	0.00	0.00	0.00	2
13	0.00	0.00	0.00	4
14	0.00	0.00	0.00	170
15	1.00	0.79	0.88	2347
16	0.00	0.00	0.00	10
17	0.50	0.60	0.54	2904
18	0.86	1.00	0.92	2145
19	0.00	0.00	0.00	1
20	0.00	0.00	0.00	716
21	1.00	0.00	0.00	722
22	0.00	0.00	0.00	16
accuracy			0.93	100777
macro avg	0.42	0.30	0.29	100777
weighted avg	0.92	0.93	0.91	100777

time: 563 ms (started: 2021-07-10 04:57:40 +01:00)

Figure 2: Classification report of principal component analysis

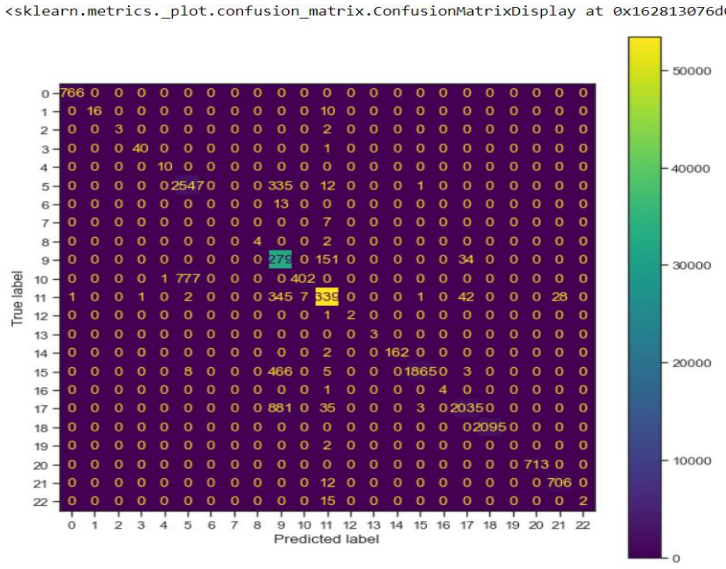

Figure 3: Confusion matrix of principal component analysis

	precision	recall	f1-score	support
0	1.00	1.00	1.00	766
1	1.00	0.62	0.76	26
2	1.00	0.60	0.75	5
3	0.98	0.98	0.98	41
4	0.91	1.00	0.95	10
5	0.76	0.88	0.82	2895
6	0.00	0.00	0.00	13
7	0.00	0.00	0.00	7
8	1.00	0.67	0.80	6
9	0.94	0.99	0.97	32982
10	0.98	0.34	0.51	1180
11	1.00	0.99	0.99	53825
12	1.00	0.67	0.80	3
13	1.00	1.00	1.00	3
14	1.00	0.99	0.99	164
15	1.00	0.79	0.88	2347
16	1.00	0.80	0.89	5
17	0.96	0.69	0.80	2954
18	1.00	1.00	1.00	2095
19	0.00	0.00	0.00	2
20	1.00	1.00	1.00	713
21	0.96	0.98	0.97	718
22	1.00	0.12	0.21	17
accuracy			0.97	100777
macro avg	0.85	0.70	0.74	100777
weighted avg	0.97	0.97	0.97	100777

Figure 4: Classification report of truncated singular value decomposition

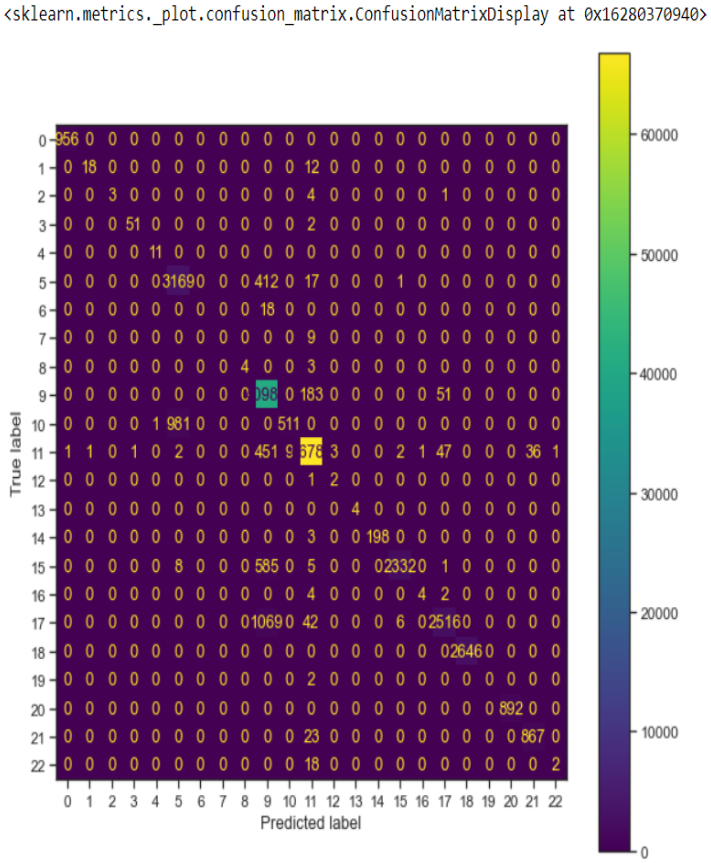

Figure 5: Confusion matrix of truncated singular value decomposition

	precision	recall	f1-score	support
0	1.00	1.00	1.00	956
1	1.00	1.00	1.00	30
2	1.00	1.00	1.00	8
3	1.00	1.00	1.00	53
4	1.00	1.00	1.00	11
5	1.00	1.00	1.00	3599
6	1.00	1.00	1.00	18
7	1.00	1.00	1.00	9
8	1.00	1.00	1.00	7
9	1.00	1.00	1.00	41214
10	1.00	1.00	1.00	1493
11	1.00	1.00	1.00	67342
12	1.00	1.00	1.00	3
13	1.00	1.00	1.00	4
14	1.00	1.00	1.00	201
15	1.00	1.00	1.00	2931
16	1.00	1.00	1.00	10
17	1.00	1.00	1.00	3633
18	1.00	1.00	1.00	2646
19	1.00	1.00	1.00	2
20	1.00	1.00	1.00	892
21	1.00	1.00	1.00	890
22	1.00	1.00	1.00	20
accuracy			1.00	125972
macro avg	1.00	1.00	1.00	125972
weighted avg	1.00	1.00	1.00	125972

time: 344 ms (started: 2021-07-10 07:18:44 +01:00)

Figure 6: Classification report for Pareto-based Monte-Carlo technique

<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x162812d6b80>

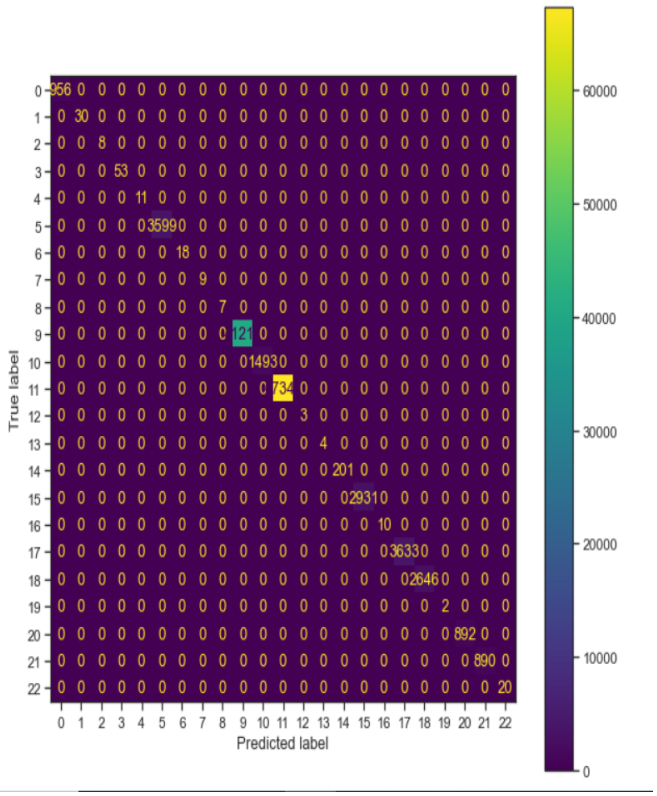

Figure 7: Confusion matrix of pareto-based Monte-carlo

Table 1 Summary of Performance metrics for the Dimensionality Reduction Techniques

Techniques	Precision	Recall	F1-score	Accuracy
Pareto	100	100	100	100
PCA	92	93	91	93
TruncatedSVD	97	97	97	97
Non- Dimensionality Reduction	83	84	82	84

Text(0.5, 1.0, 'Training Model Performance')

Figure 8: Model Evaluation in Terms of Performance

V. DISCUSSION OF RESULTS

From the results obtained, in figure 2, the classification report for the support vector classifier on PCA shows 92% precision and 93% accuracy. The confusion matrix plotted in other to show which of the techniques gave a true prediction and shown in figures 3, 5, and 7 indicate that the PCA gave a correct prediction of label

339 times (figure 5). Also, figure 4 showed that the classification report from SVM on TruncatedSVD produced 97% precision and 97% accuracy; while the confusion matrix in figure 5 showed that TruncatedSVD predicted a label as true 678 times. Figure 6 shows the classification report of PB-MCFR which clearly indicate both precision and accuracy results of 100% while it predicted the same label as true for 734 times as shown in the confusion matrix in figure 7. The performance evaluation of the three dimensionality reduction techniques in terms of training performance and the reduction in false positives, and false negatives is shown in table 1, indicating that the PB-MCFR technique has the highest training performance of about 100% of accuracy, followed by TruncatedSVD with an accuracy of about 97% and lastly, PCA which had an accuracy of about 93%. The dataset was also trained using all the features including the irrelevant features. This was to show the impact of not applying dimensionality reduction on the dataset. The accuracy shown in table 1 indicate that the dataset without dimensionality reduction results to a low accuracy of 84%. Figure 8 shows a graph representation of the performance evaluation of the techniques used. The accuracy values is plotted on the y-axis and the dimensionality techniques used is plotted on the x-axis. From the graph, it is visible that Pareto case produces the highest performance when it was trained with the machine learning model; whereas when the dataset is trained with all the features, that is, no dimensionality reduction on the data, it produces a low accuracy. This shows that Pareto-Based Monte Carlo Technique is suitable for reduction in dimensionality for efficient models for the detection of zero-day attacks.

VI. CONCLUSION

This paper compared three dimensionality reduction techniques on a dataset for zero-day attack. The comparison was carried out in terms of classification report (which shows the percentage of accuracy, precision and recall), confusion matrix (which shows the true predicted level vs the original label data). The reduction techniques of dimensionality used on this work include principal component analysis (PCA), truncated singular value decomposition (TrncatedSVD) technique and pareto-based Monte carlo (PB-MCFR) technique. For the comparison of which techniques is more visible on a zero day attack (KDD) dataset, PB-MCFR technique is more visible than that of PCA and truncateSVD. For accuracy and correctness in prediction, the PB-MCFR technique was higher than that of principal component analysis and truncated singular value decomposition with an accuracy of 100%. The full dataset withour any dimensionality reduction was also used to train the model and was observed to produce a low accuracy when compared to applying dimensionality reduction. The analysis showed that dimensionality reduction is important in a dataset and that the pareto-based Monte carlo technique is more efficient for dimensionality reduction on a zero day attack dataset.

REFERENCES

[1] AlEroud, A., & Karabatis, G. (2012). A contextual anomaly detection approach to discover zero-day attacks. *Paper presented at the 2012 International Conference on Cyber Security*. 40-45.

[2] Chumachenko, K. (2017). Machine Learning Methods for Malware Detection and Classification.

[3] Comar, P. M., Liu, L., Saha, S., Tan, P.-N., & Nucci, A. (2013). Combining supervised and unsupervised learning for zero-day malware detection. *Paper presented at the 2013 Proceedings IEEE INFOCOM*. 2022-2030.

[4] Dalziel, H. (2014). How to defeat advanced malware: new tools for protection and forensics: *Syngress*.

[5] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. In: MIT Press.

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

[7] Kaur, R., & Singh, M. (2014). Efficient hybrid technique for detecting zero-day polymorphic worms. *Paper presented at the 2014 IEEE International Advance Computing Conference (IACC)*.

[8] Kaur, R., & Singh, M. (2015). A Hybrid Real-Time Zero-day Attack Detection and Analysis System. *International Journal of Computer Network and Information Security, 9*, 19-31. doi: 10.5815/ijcnis.2015.09.03

[9] Parrend, P., Navarro, J., Guigou, F., Deruyver, A., & Collet, P. (2018). Foundations and applications of artificial Intelligence for zero-day and Monte-step attack detection. *Journal on Information Security EURASIP, 2018*(1), 4.

[10] Portokalidis, G., & Bos, H. (2007). SweetBait: Zero-hour worm detection and containment using low-and high-interaction honeypots. *Journal of Computer Networks, 51*(5), 1256-1274.

[11] Reddy, G. T., Reddy, M. P., Lakshmanna, L., Kaluri, R., Rajput, D. S., Srivastava, G., & Baker, T. (2020). Analysis of Dimensionality Reduction Techniques on Big Data. *IEEE Access, 8, 54776-54788.*

[12] Singh, U. K., Joshi, C., & Kanellopoulos, D. (2019). A framework for zero-day vulnerabilities detection and prioritization. *Journal of Information Security and Applications, 46*, 164-172.

[13] Singh, U. K., & Joshi, C. (2018). Scalable approach towards discovery of unknown vulnerabilities. *International Journal of Network Security*. 20(5), 827-835.

[14] Singh, U. K., Joshi, C., & Singh, S. K. (2017). Zero day attacks defense technique for protecting system against unknown vulnerabilities. *International Journal of Scientific Research in Computer Science and Engineering. 5*(1), 13-18.

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

[16] Song, J., Takakura, H., & Okabe, Y. (2008). Cooperation of intelligent honeypots to detect unknown malicious codes. *Paper presented at the 2008 WOMBAT Workshop on Information Security Threats Data Collection and Sharing.*