Journal of Information Systems and Informatics
Vol. 5, No. 3, September 2023 e-ISSN: 2656-4882 p-ISSN: 2656-5935
DOI: 10.51519/journalisi.v5i3.546 Published By DRPM-UBD

Securing Against Zero-Day Attacks: A Machine Learning Approach for Classification and Organizations' Perception of its Impact

Anietie P. Ekong[1], Aniebiet Etuk[2], Saviour Inyang[3], Mary Ekere-obong[4]

[1,2,3,4]Department of Computer Science, Akwa Ibom State University, Ikot Akpaden, Nigeria
Email: anietieekong@aksu.edu.ng[1], bietsoft@gmail.com[2], sinyang@yahoo.com[3], maryobong@gmail.com[4]

Abstract

Zero-day malware is a type of malware that exploits system vulnerabilities before it is detected and sealed. This type of malware is a significant threat to enterprise cybersecurity and has tremendous impact on organizations' performance, as it can spread widely before organizations can clamp down on the threat. Unfortunately, exploit developers can attack system's vulnerabilities at a pace that is faster than defensive patches. In this research, classification of zero-day attack was carried out. Exploratory Data Analysis (EDA) on malware zero data was conducted. Then feature selection was carried out using Principal Component Analysis (PCA) for the selection of the most important features in the dataset after which a Random Forest (RF) Algorithm was adopted for the classification of zero-day attack. The impact of such attacks was also analyzed, and results were evaluated using confusion matrix and an accuracy of 95% in the classification of zero-day attack with a class error of 3.8% was obtained. A survey of the perception of the potential impacts of these attacks on organization was also carried out. These results indicate efficiency of machine learning algorithm in the classification of attacks as zero-day malware attacks or not. The research also offered pragmatic insights into the perception by organizations of its potential negative impacts and their eagerness to embrace and prioritize proffered cyber security solution(s) to avoid such attacks in order to avert undesirable consequences.

Keywords: Zero-day Attack, Machine Learning, Organization Data security, Cyber security, Random Forest.

1. INTRODUCTION

A zero-day (0-day) vulnerability is a loophole in a software that is latent. A zero-day exploit is the way a vulnerable system is attacked [2]. These are threats with a higher probability of succeeding since organizations are likely not to have complete and appropriate measures in place to nip them in the bud. A zero-day attack is termed as such since it occurs before the target knows about the existence of such vulnerability as the malware is released before or with little opportunity for developers to patch any existent vulnerability [1]. The most dangerous aspect of a zero-day attack is that the application having vulnerability may be dealing with

sensitive files [6]. If such an application is on a server, an attack can pass through it and hijack the whole system. A lot of losses have happened over the years because of this attack [8]. Organizations have run into several problems which include but not limited to revenue loss, legal repercussions, reputational damage, loss/theft of data, decrease in production and unauthorized access. Service quality has a major impact on clients trust on organizations [5]. Zero attacks can lead to loss of such trust. Machine Learning, being the ability of a system to learn from experience is indispensable in discovering patterns in data and these can be rightly applied to classify attacks as zero day or not.

In recent times, researchers have hunted for practical solutions to at least remedy the situation immediately the attack happens [13]. This attack cannot be discovered until it happens, that is the most dreaded part of it. One best way to prevent this kind of cyber-attack is to purchase software products that have passed through series of tests and confirmations from trusted software and programmers should ensure every loophole is sealed in order to avoid attackers from exploiting any vulnerable loophole in an application [9]. People who have cloud storage and use applications or API to read and process data should ensure there is maximum testing for those cloud applications. Zero-day attack is always quick and zero-day attackers don't lose their fight easily [6]. Machine learning algorithms ensure that classification is correctly and timeously done [3]. The zero-day threats (ZDTs) to organizations' networks is costly and require new approaches to identify malicious behavior [2].

Many studies have been conducted about zero-day attacks. One of the studies was Zero Day Threat Detection Using Metric Learning Autoencoders which demonstrated and improved upon a previous approach which used a dual-autoencoder to identify such threats in the network flow [2]. Further studies were carried out using principal component analysis (PCA), truncated singular value decomposition technique (TruncatedSVD) and pareto-based Monte carlo technique (PB-MCFR) for dimensionality reduction on a dataset for zero-day vulnerability analysis. The results showed that PB-MCFR outperformed PCA and TruncatedSVD and concluded that while significant efforts have been put in developing a robust tool to combat zero-day attacks, they fall short in the key performance metrics [12]. Also, machine learning algorithms were to see how well ML algorithms can detect zero-day malware, with random forest showing the best result in terms of accuracy with zero rates for false positive and false negative [6]. Also, in evaluating AI-Based Techniques for Zero-Day Attacks Detection using high-level model abstractions and non-linear transformations, it was observed that due to the sensitivity of the zero-day attacks, accuracy or precision was not enough to measure the performance of the models, as it is an ever-evolving issue, the use of current datasets should be employed [10].

An Enhanced Classification Model for Likelihood of Zero-Day Attack Detection and Estimation based on deep-reinforcement learning, a reward learning and training feature with sparse feature generation and adaptive multi-layered recurrent approach that performed better than rule-based ranking in predicting zero-day threats was used [11]. In another development, a transferred generative adversarial network(tDCGAN) based on deep auto encoders for detection of malware was used and an attempt was made to solve the problems by malware by using the discriminator's ability to extract meaningful features for malware detection and an accuracy of 95.74% was achieved [7].

All the reviewed work did not address the negative impact of this attack on organization neither did they get the perception of such impact on organizations in addition to insufficiency of the metrics for the classification. It is to this end that there is a need to build a model that helps organizations classify attacks in order to identify and seal such vulnerabilities and a contemporary approach, Random Forest, a machine learning algorithm, with its excellence ability to handle intricate datasets and still classify with high accuracy offers one of the best approaches in this direction hence its adoption in this research.

2. METHODS

This research seeks to classify malware Zero-day Attack using Machine Learning approach. The data includes both malicious as well as legitimate files that were sourced from Meraz'18, the annual techno-cultural festival held at IIT Bhilai. The malware section of the data constitutes software that is intended to interfere, harm, or acquire unauthorized entry to a computer's infrastructure while the legitimate files section of the data were programs that are safe for users to utilize and devoid of malicious intent [14].

Furthermore, Algorithm Based Method (ABM) was adopted in this research. In various fields, notably computer science, engineering, and psychology, the application of algorithmic methods is a common strategy for addressing or solving problems [15]. Using ABM, we itemize the process of solving the problem of classification of zero-day attack by developing the algorithm for the method in section 2.1 which outlines the process from the point of model/problem definition, data acquisition, carrying out EDA, feature selection, data training using Random Forest classifier and evaluating the result of the classifier.

2.1 Algorithm for The Method

```
1: Begin
2: Model / problem Definition
2: Get Malware data from [14];
3: Carryout EDA;
4: Carryout Feature Selection;
5: From 4:, Identify important features;
6: For i=1 to No_of_Features {
7: feature = colnames(pc_loadings), importance=abs(pc_loadings)*
                pc_variance+abs(pc_loadings) * pc_variance;
9: i++; }
10: Plot pc in 7;
11: Store data in dataframe;
12: Perform normalization from 11;
13: Split the normalized data in 11 in the Ratio 70:30;
14: Apply ML model to 70% of the data in 13;
15: Test the model with the remaining 30% of data in 13
16: Evaluate Result of 15
17: End
```

2.2 Model / Problem Definition

A software fragility known as a zero-day vulnerability is one for which no recognized security patch or upgrade has been made available. There is no publicly accessible data about this danger, and a software provider may or may not be aware of the vulnerability. Nevertheless, this study aim to use machine learning algorithm in the classification of zero-day malware attacks using Random Forest Classifier and using Principal Component analysis for the feature selection process. First in this research, a dataset consisting of $d + 1$ dimensions is used. Furthermore, the mean and dimension of every section in the dataset are computed and the computation of the covariance matrix is done in accordance with Eqn 1.

$$CO(x, y) = \frac{1}{n}\sum_{i=1}^{n}(x - \bar{x})\ (y - \bar{y}) \quad (1)$$

Furthermore, the computation and sorting of eigenvectors will be carried out. Finally, the data is transformed into the new subspace using this $d * 1$ eigenvector matrix. After the feature selection process, the machine learning algorithm, RF, is applied. The reason for using this algorithm is that in detecting and classifying potential risks or malicious activities, Random Forest is recognized for its exceptional predictive accuracy. Being an ensemble learning method, it combines multiple decision trees to generate forecasts. This ensemble approach contributes to improving the overall efficiency and reliability of the classification model, a critical factor in cybersecurity for accurate threat identification and mitigation. Also, in the realm of cybersecurity, Random Forest is indispensable because it

excels at diminishing overfitting, managing intricate datasets, operating effectively with extensive databases, and delivering precise predictions with high accuracy [16] The random forest algorithm utilizes bagging on decision trees but with a crucial enhancement. In training the model, the following steps apply.

1. Take a bootstrap (with replacement) subsample from the data.
2. For the first split, sample $p < P$ variables at random without replacement.
3. For each of the variables of the dataset $X_{i_1} \ldots X_{j(p)}$, apply the splitting algorithm:
4. For each split values $s\_j(k)$ of $X_{j(k)}$:
5. Split the data in partition A, with $X_{j(k)} < s_{j(k)}$ as one partition and the rest of the data where $X_{j(k)} \geq s_{j(k)}$ as some other partition.
6. Measure the homogeneity of classes within the partitions of A.
7. Select the value of $s_{j(k)}$ that produces the split value $s_{j(k)}$that gives the maximum within partition homogeneity of class.
8. Select the variable $X_{j(k)}$ and split the values $s_{j(k)}$ that produces maximum within partition homogeneity of class.
9. Proceed to the next split and repeat from step 2
10. 10: Continue with additional splits following the same procedure until the tree is grown.

2.2 Exploratory Data Analysis (EDA)

Exploratory data analysis is the crucial process of conducting early investigations on data to find patterns, identify anomalies, test hypotheses, and double-check assumptions using summaries of statistics and graphical representations [13]. Understanding the data first and attempting to extract as many ideas from it as possible is good practice. EDA is all about interpreting the data at hand before using it. Prior to conducting data analysis and passing it through an algorithm, it is essential to thoroughly comprehend it. Patterns in the data were recognized and the decision on which factors are crucial and the ones that has little bearing on the result was made. Every machine learning problem solving involves EDA. In this section, a cross section of variables in the Zero-day malware attack dataset was obtained from [14] and different visualization of the statistical measures of the different parameters are presented. Figure1 shows the workflow of the research which are series of macro steps employed. Also since there was more features in the dataset, PCA for feature selection was employed. After the feature selection, Random Forest for the classification of the reduced feature data sets was done and an evaluation of the classifier in order to determine the accuracy of the model was done.

Furthermore, figure 2 represents a major linker version count in the datasets, it shows that versions from 1 to 15 have a greater number of counts in the data. With respect to data, the major linker version signifies the specific iteration of the linker software employed to unite object files, amalgamating them into an executable or library file. This integration process, orchestrated by a computer application termed the linker, merges numerous object files into one, be it an executable or library file.

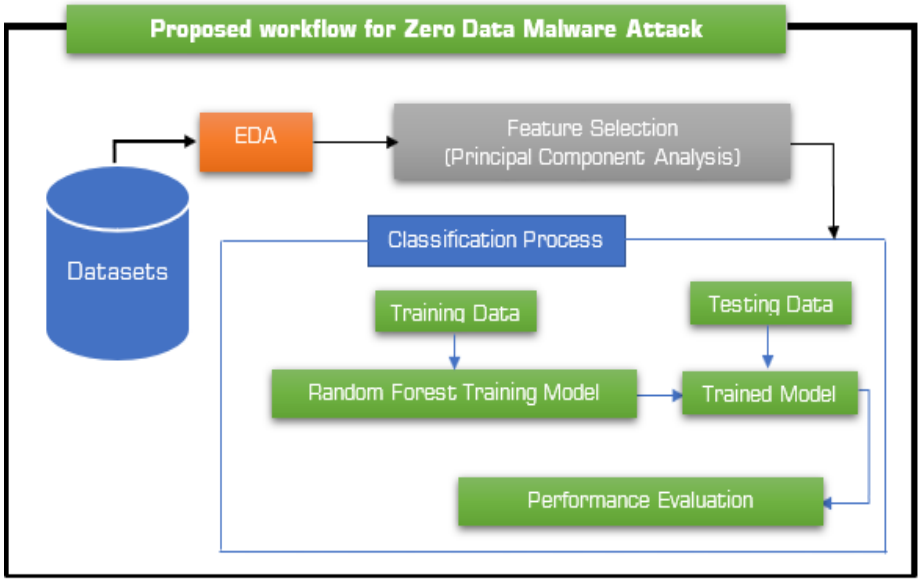

Figure 1. Proposed Workflow

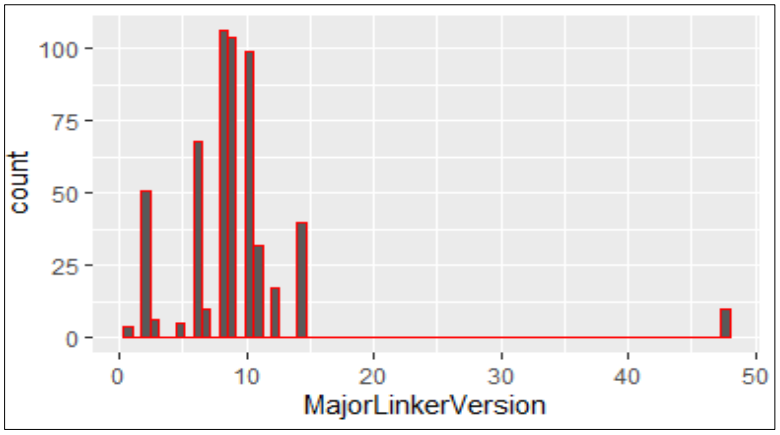

Figure 2. MajorLinkerVersion vs Count

The concept of a minor linker Version in figure 3 pertains to the software iteration of the linker application, which is employed for the amalgamation of object files to generate executable or library files. This linker version data is used to monitor adjustments and enhancements to the linker program and to ensure its compatibility with other software elements. From figure 3, a minor linker version count in the datasets shows that version between 0 to 25 has a greater number of counts in the data.

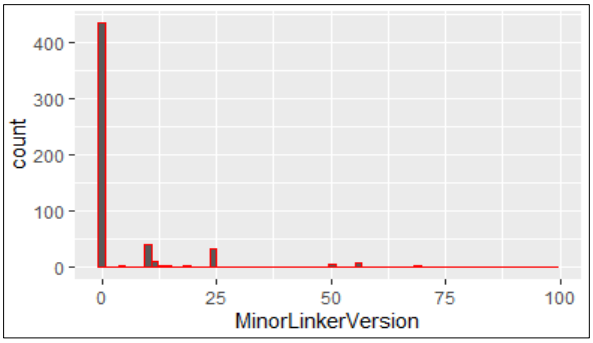

Figure3. MajorLinkerVersion vs Count

Furthermore, File alignment describes the procedure of matching the data or portions of a file to particular positions or addresses. It guarantees that the beginning memory address of each section is a multiple of the alignment value. Figure 4 depicts the file alignment count in the data sets.

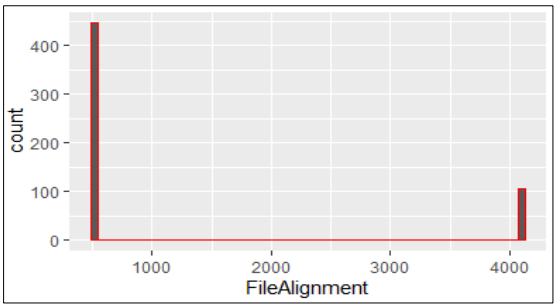

Figure 4. FileAlignment vs Count

Again, Section alignment makes sure that each section begins at a memory address that is double the alignment value in order to improve the initialization and operation of the executable file. A total of 4000 had the maximum number of counts in the data set as compared to the 8000 alignments which had less than 100 counts while sections alignments of 5000, 6000 and 7000 did not have any count in the data sets. Figure 5 depicts section alignment vs count in the dataset.

In the context of cyber security, entropy typically refers to a measurement of data's randomization or unpredictably. While low entropy denotes patterns or frequency, which can be more easily exploited or attacked, high entropy indicates a higher degree of unpredictability, making data more challenging to anticipate or analyze.

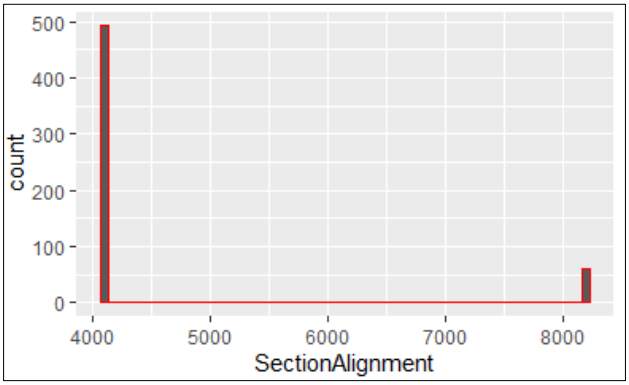

Figure 5. FileAlignment vs Count

Figure 6 and Figure 7 show the maximum and minimum entropy respectively in the dataset. The maximum and minimum entropy indicates the maximum and minimum degree of disorderliness within the dataset. They are measures of pinpointing potential attack pathways and uncovering weaknesses and identifying attacks during analysis in cybersecurity related events. For example, it can be observed that the entropy of 3.7 has the maximum number of counts in the dataset. Figure 8 depicts the load configuration size as the program settings are loaded from a configuration file by the Load Configuration command.

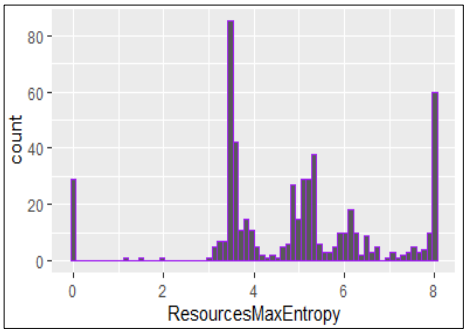

Figure 6. Resource Max Entropy vs Count

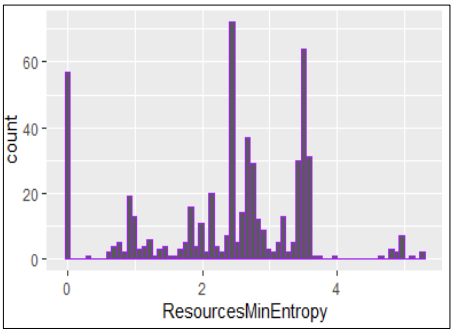

Figure 7. Resource Min Entropy vs Count

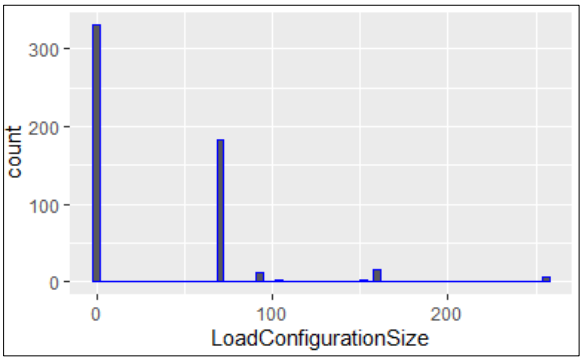

Figure 8. Resource Min Entropy vs Count

Moreso, figure 9 represents the image base count in the data set. It's crucial to keep in mind that the "ImageBase" notion is unique to the Windows file format and could not work with other operating systems or executable file formats.
Recognizing the ImageBase and how it relates to The Address Space Randomization and memory layout is essential when performing software security analysis in order to spot possible vectors of attack and vulnerabilities in software applications.

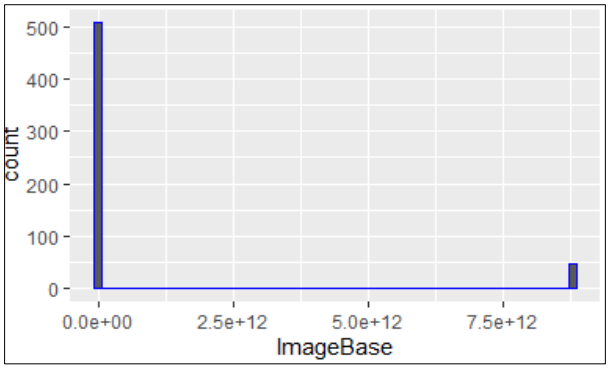

Figure 9. Resource Max Entropy vs Count

Figure 10 and figure 11 represent various parameters in the datasets that is used to visualized how each of the indepentent variables contributes to the class label of the factor variable (i.e. illegitimate 0 or legitimate 1). From Figure 10, being a graph of Resource Mean Entropy vs Legitimate, it can be observed that resoure mean entropy is more illegitimate i.e. 0 when compared with legitimate i.e. 1. Also, Figure 11 which shows Section Mean Entropy vs Legitimate represents a box plot that visualized how section mean entropy fits between the two class label, it could be observed that both class labels tend to have even distributions in the class of 0 and 1 which represents illegitimates and legitimates respectively.

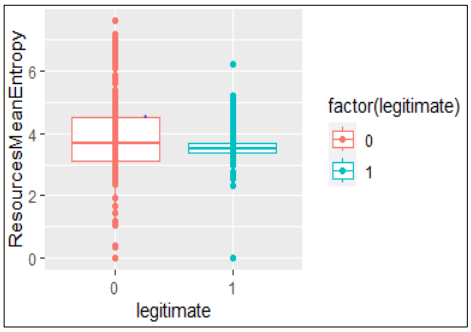

Figure 10. Resource Mean Entropy vs Legitimate

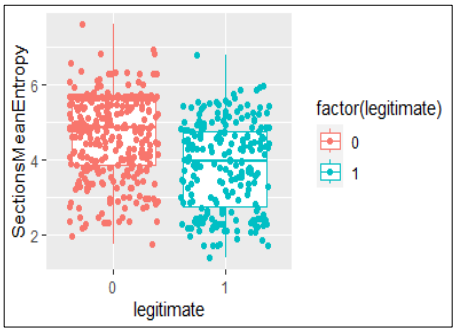

Figure 11. Section Mean Entropy vs Legitimate

2.3 Feature Selection

The process of choosing a subset of pertinent features (variables, predictors) for use in model building in machine learning and statistics is known as feature selection, variable selection, attribute selection, or variable subset selection. One of the key elements of a feature engineering method is the feature selection process. A predictive model is created after lowering the amount of input variables. Measuring how important a chosen feature is, choosing the best is the key. In this research, PCA was adopted for feature selection because in datasets containing numerous attributes as used here, it can resolve issues related to interdependent characteristics, a common challenge encountered by other feature selection methods. The algorithm for the selection using PCA is as presented in the ensuing paragraph.

1. Load numerical parameters in the datasets.
2. Perform PCA by scaling the data to have a mean of 0 and a unit variance.
3. Extract the loadings and variance explained by each principal component (PC).
4. Ensure The rotation attribute of the PCA object contains the loadings, or the coefficients that define the PCs. The sdev attribute has the standard deviation of each principal component.
5. Calculate a measure of importance for each feature based on the loadings and variance explained. Specifically, take the absolute value of the loadings for the first two PCs and multiply them by the corresponding variance to get a weighted measure of importance for each feature and then store this information in a data frame and sort it in descending order of importance.

The normalized PC features in the data sets alongside its importance are in table 1 and the weighted measure of importance is illustrated in figure 12. It can be

observed from table 1 that PC43, PC31, PC18, PC30, PC44 and PC 23 respectively were least significant components in the features of the datasets as compared to other components.

Table 1. Principal Components

Variables	Feature	Importance
SectionsMeanRawsize	PC35	0.049720734
ResourcesMaxSize	PC51	0.048835019
SectionsMinVirtualsize	PC39	0.048065995
SectionsMinRawsize	PC36	0.047938213
ResourcesMeanSize	PC49	0.045645147
SectionsMeanVirtualsize	PC38	0.043597577
SectionMaxRawsize	PC37	0.038134134
BaseOfData	PC11	0.037779796
MajorSubsystemVersion	PC19	0.03565149
AddressOfEntryPoint	PC9	0.03342814
MajorOperatingSystemVersion	PC15	0.032796943
SizeOfOptionalHeader	PC2	0.032429612
Machine	PC1	0.032429612
SizeOfHeaders	PC22	0.031021614
ImageBase	PC12	0.029451922
SizeOfUninitializedData	PC8	0.029237422
BaseOfCode	PC10	0.029139717
SizeOfImage	PC21	0.027937062
SectionsMeanEntropy	PC32	0.026639474
SectionMaxVirtualsize	PC40	0.026334989
FileAlignment	PC14	0.026255617
legitimate	PC54	0.025376194
SectionsMinEntropy	PC33	0.025027066
SizeOfInitializedData	PC7	0.024926425
SizeOfCode	PC6	0.022409389
ResourcesMinEntropy	PC47	0.020943967
Subsystem	PC24	0.020738505
LoadConfigurationSize	PC52	0.01966658
MinorLinkerVersion	PC5	0.019606593
MinorSubsystemVersion	PC20	0.018624637
MinorOperatingSystemVersion	PC16	0.01722436
ImportsNbDLL	PC41	0.015789812
ResourcesMaxEntropy	PC48	0.014848786
SectionAlignment	PC13	0.014337533
MajorLinkerVersion	PC4	0.013578145
SectionsMaxEntropy	PC34	0.012295237

Variables	Feature	Importance
VersionInformationSize	PC53	0.011571535
ImportsNb	PC42	0.011341961
SizeOfStackReserve	PC26	0.009156378
ResourcesMeanEntropy	PC46	0.008332446
ResourcesNb	PC45	0.00692906
ResourcesMinSize	PC50	0.005188555
ImportsNbOrdinal	PC43	0.004722707
Characteristics	PC3	0.004277612
SectionsNb	PC31	0.004160642
SizeOfHeapCommit	PC29	0.00344689
MinorImageVersion	PC18	0.0031701
MajorImageVersion	PC17	0.003080579
LoaderFlags	PC30	0.002913406
DllCharacteristics	PC25	0.002456751
ExportNb	PC44	0.002334179
SizeOfHeapReserve	PC28	0.002173655
CheckSum	PC23	0.001823451
SizeOfStackCommit	PC27	0.00140238

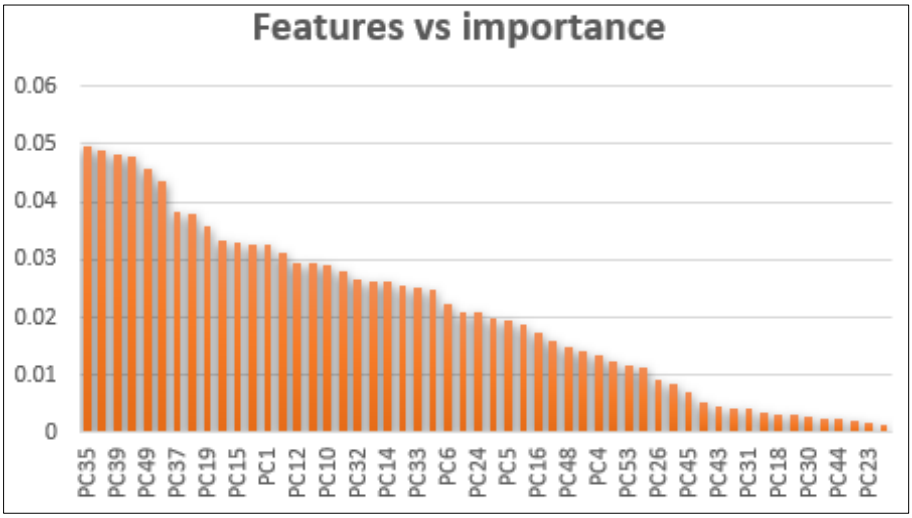

Figure 11. Feature vs Importance

Finally, to find out how organizations feel about the threat of a zero-day attack, a survey of 20 organizations was carried out to find out their perception of having such attacks. The survey was carried out in Akwa Ibom State, Nigeria using questionnaire. Though 15 of respondents said they had never heard of it, all of them perceived it as a potentially serious threat that deserved serious attention and

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

as an attack that could ground their organizations and could potentially lead to job loss, litigations, low productivity, breach or organization privacy and even outright business closure among other undesirable and unimagined consequences and were interested in implementing cyber security measures and prioritizing them henceforth to prevent such attacks provided it is within affordable limits.

3. RESULTS AND DISCUSSION

In assessing the effectiveness, efficiency, impact of this research in line with the aim, an evaluation process using R programming to preprocess all needed variables to obtain a summary of all parameters based on min, 1st Qu., Median, mean,3rd Qu and max for each of the features in the data sets. Figure 12 indicates the statistical tendencies for these variables.

```
    Machine      SizeOfOptionalHeader Characteristics MajorLinkerVersion MinorLinkerVersion   SizeOfCode
Min.   :  332   Min.   :224.0        Min.   :   14   Min.   : 1.000     Min.   : 0.000     Min.   :     0
1st Qu.:  332   1st Qu.:224.0        1st Qu.:  258   1st Qu.: 7.000     1st Qu.: 0.000     1st Qu.:  24576
Median :  332   Median :224.0        Median :  271   Median : 9.000     Median : 0.000     Median :  88064
Mean   : 4344   Mean   :225.9        Mean   : 4934   Mean   : 9.053     Mean   : 4.728     Mean   : 235997
3rd Qu.:  332   3rd Qu.:224.0        3rd Qu.: 8450   3rd Qu.:10.000     3rd Qu.: 0.000     3rd Qu.: 189440
Max.   :34404   Max.   :240.0        Max.   :33679   Max.   :48.000     Max.   :99.000     Max.   :3874816
SizeOfInitializedData SizeOfUninitializedData AddressOfEntryPoint   BaseOfCode         BaseOfData
Min.   :       0      Min.   :       0        Min.   :       0    Min.   :       0   Min.   :       0
1st Qu.:    9216      1st Qu.:       0        1st Qu.:   13703    1st Qu.:    4096   1st Qu.:   16384
Median :   57344      Median :       0        Median :   54856    Median :    4096   Median :   67584
Mean   :  349082      Mean   :   37142        Mean   :  301180    Mean   :   38816   Mean   :  242384
3rd Qu.:  382464      3rd Qu.:       0        3rd Qu.:  170892    3rd Qu.:    4096   3rd Qu.:  167936
Max.   :58178560      Max.   :10256384        Max.   :12627968    Max.   :10260480   Max.   :12197888
  ImageBase         SectionAlignment FileAlignment  MajorOperatingSystemVersion MinorOperatingSystemVersion
Min.   :6.554e+04   Min.   :4096     Min.   : 512   Min.   : 1.000              Min.   : 0.0000
1st Qu.:4.194e+06   1st Qu.:4096     1st Qu.: 512   1st Qu.: 4.000              1st Qu.: 0.0000
Median :4.194e+06   Median :4096     Median : 512   Median : 5.000              Median : 0.0000
Mean   :7.328e+11   Mean   :4541     Mean   :1200   Mean   : 4.848              Mean   : 0.5145
3rd Qu.:3.062e+08   3rd Qu.:4096     3rd Qu.: 512   3rd Qu.: 5.000              3rd Qu.: 1.0000
Max.   :8.790e+12   Max.   :8192     Max.   :4096   Max.   :10.000              Max.   :11.0000
MajorImageVersion  MinorImageVersion  MajorSubsystemVersion MinorSubsystemVersion  SizeOfImage       SizeOfHeaders
Min.   :    0.00   Min.   :    0.00   Min.   : 3.000        Min.   : 0.0000       Min.   :    8192   Min.   : 512
1st Qu.:    0.00   1st Qu.:    0.00   1st Qu.: 4.000        1st Qu.: 0.0000       1st Qu.:   98304   1st Qu.:1024
Median :    0.00   Median :    0.00   Median : 5.000        Median : 0.0000       Median :  358400   Median :1024
Mean   :   83.37   Mean   :   74.68   Mean   : 4.904        Mean   : 0.5199       Mean   :  842805   Mean   :1683
3rd Qu.:    6.00   3rd Qu.:    0.00   3rd Qu.: 5.000        3rd Qu.: 1.0000       3rd Qu.:  714752   3rd Qu.:1024
Max.   :21315.00   Max.   :20512.00   Max.   :10.000        Max.   :20.0000       Max.   :58343424   Max.   :4096
   CheckSum          Subsystem     DllCharacteristics SizeOfStackReserve SizeOfStackCommit SizeOfHeapReserve
Min.   :0.000e+00   Min.   :1.000   Min.   :    0      Min.   :       0   Min.   :      0   Min.   :      0
1st Qu.:0.000e+00   1st Qu.:2.000   1st Qu.:    0      1st Qu.:  262144   1st Qu.:   4096   1st Qu.:1048576
Median :1.506e+05   Median :2.000   Median : 1344      Median : 1048576   Median :   4096   Median :1048576
Mean   :7.235e+07   Mean   :2.319   Mean   :15919      Mean   : 1075646   Mean   :   7413   Mean   :1021076
3rd Qu.:5.503e+05   3rd Qu.:3.000   3rd Qu.:33088      3rd Qu.: 1048576   3rd Qu.:   4096   3rd Qu.:1048576
Max.   :2.814e+09   Max.   :9.000   Max.   :49504      Max.   :33554432   Max.   :1200128   Max.   :4194304
```

Figure 12. statistical tendencies for variables

The data underwent preprocessing to remove noisy and redundant data. The data was then split into 70% training set and 30% test set for the Machine Learning model. Figure 13 is a snapshot of the training sets.

Journal of Information Systems and Informatics
Vol. 5, No. 3, September 2023
p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

	Machine	SizeOfOptionalHeader	Characteristics	MajorLinkerVersion	MinorLinkerVersion	SizeOfCode	SizeOfInitializedData	SizeOfUninitializedData
2	332	224	258	9	0	84480	25600	
3	332	224	8450	8	0	4608	3584	
4	332	224	8450	10	0	108544	15872	
5	332	224	8226	48	0	513024	2048	
6	332	224	8450	8	0	385024	769024	
8	34404	240	8226	10	0	74240	77312	
12	332	224	8450	9	0	238080	29696	
13	332	224	8450	8	0	131584	37376	
16	332	224	258	9	0	34304	6656	
17	332	224	8226	48	0	5632	2560	
18	332	224	8450	10	10	60416	15360	
19	332	224	8450	8	0	62464	589312	

Showing 1 to 12 of 391 entries, 55 total columns

Figure 13. Cross section of training Sets

The training dataset is fed into the Random Forest algorithm to create a training model that can be used to classify new or different datasets in R. The outcome is as shown in Figure 14.

```
                Type of random forest: classification
                      Number of trees: 601
No. of variables tried at each split: 3

        OOB estimate of  error rate: 3.84%
Confusion matrix:
    0   1 class.error
0 228   7  0.02978723
1   8 148  0.05128205
```

Figure 14. Outcome of the Training Set

Furthermore, the diagrammatic representation of the random forest tree of the model built is as shown in figure which shows the number of nodes in the random forest models being represented in a hierarchical tree indicating the classes (legitimate and non-legitimate zero-day attack i.e., 1 and 0).

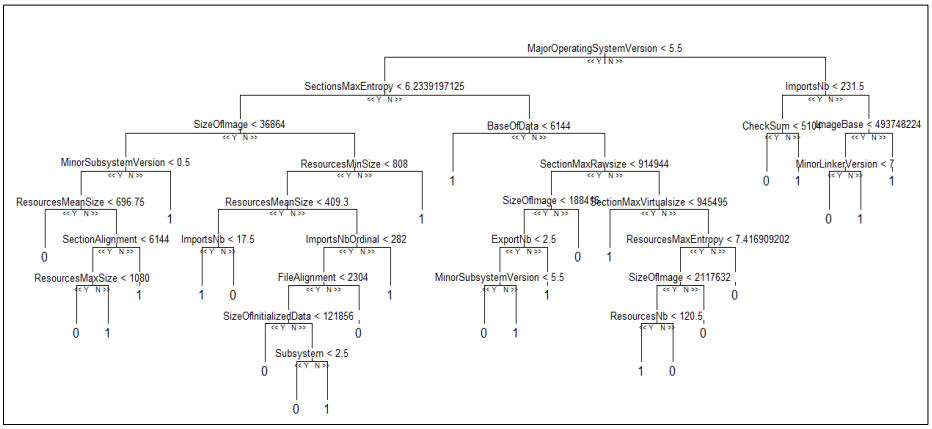

Figure 15. Random Forest for the model

From Figure 15, the errors associated with the generated tree must be identified. This is done by plotting errors against each node of the tree as shown in figure 16. It can rightly be observed that as the number of trees keeps on increasing, the error keeps decreasing which shows that using a larger number of trees in the Random Forest model will give a better accuracy.

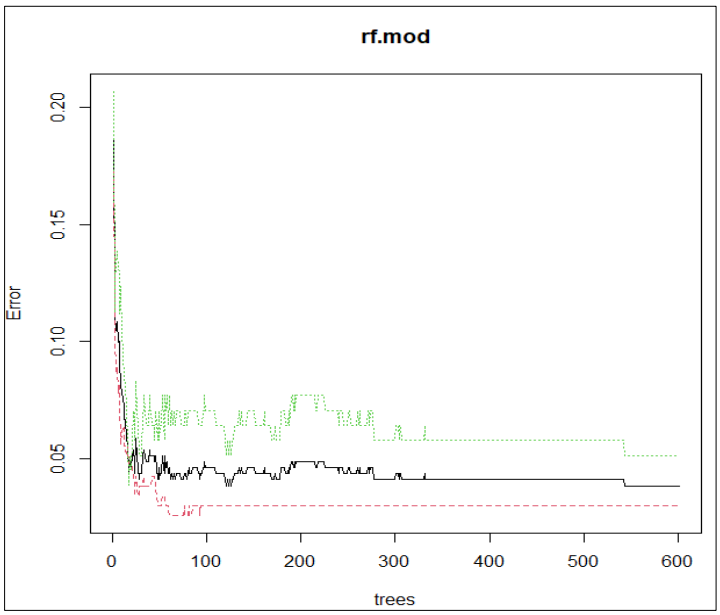

Figure 15. Error on the rf tress.

Furthermore, the confusion matrix is adopted for evaluation of the classification of the model. The confusion matrix as expressed in table 2 is specifically used for the evaluation of the accuracy of the model. The test dataset which was use in the testing of the random forest model shows that out of 104 illegitimate files, 99 were correctly classified as illegitimate files while 5 were miss-classified as legitimate. Also, for the legitimate class, out of 57 legitimates files, 54 were correctly classified as legitimate files while 3 were miss classified as illegitimate files. From the confusion matrix, using sum(diag(t)/sum(t)) =((99+54)/(99+5+3+54)) *100 gives an accuracy of 0.95 i.e. 95%.

Table 2. Confusion matrix

Prediction	**Actual**	
	0	1
0	99	5
1	3	54

4. CONCLUSION

Software vendors and organizations need to continuously check for new vulnerabilities in their software and prepare for zero-day attacks. While it is difficult to completely prevent it, several defensive measures can be taken to protect against this menace can be taken. Everyday strategies like antivirus, spywares etc are not enough to identify these and previous work on this, apart from using few metrices for classification, did not address the perceptions of organizations with respect to its negative consequences. This study delved into the realm of zero-day malware, a perilous form of cyber threat exploiting system vulnerabilities before detection and remedy to bridge this gap as such attacks are of severe risks to enterprise security, wreaking havoc on organizational efficiency as it proliferates undetected. This study carves a distinctive path by focusing on the classification of zero-day attacks and particularly important is its embracement of machine learning in classification, a potent asset in the cyber security world. The employment of ensemble machine learning approach for this classification is innovative. The choice of the Random Forest Algorithm is especially noteworthy for its record of delivering precise outcomes using intricate datasets. Classifiers were trained to detect known classes and adapt to new ones. This algorithm was employed to properly classify attacks as zero-day or not and this yielded commendable results of 95% accuracy and 3.8% error rate. This research not only advances the comprehension of zero-day attacks but also offers pragmatic insights into the perception by organizations, of its impact and their eagerness to embrace and prioritize any proffered solution(s). Looking ahead, it is crucial to acknowledge the ever-evolving nature of zero-day threats. Recognizing the necessity for organization-specific datasets as pivotal since security configurations and practices can widely differ among organizations and such diversities should be explored. Comparative analyses of these datasets could illuminate disparities in patterns, shedding light on how distinct security measures influence the occurrence and characteristics of these attacks. Overall, the research aim was met.

REFERENCES

[1] D. Nandakumar, R. Schiller, C. Redino, K. Choi, A. Rahman, E. Bowen, M. Vucovich, M. Weeks, and A. Shaha, "Zero Day Threat Detection Using Metric Learning Autoencoders," *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*, Nassau, Bahamas, 2022, pp. 1318-1325, https://doi: 10.1109/ICMLA55696.2022.00210.

[2] A. Goldsteen, G. Ezov, R. Shmelkin, M. Moffie, and A. Farkash, "Anonymizing machine learning models," in *International Workshop on Data Privacy Management*, Oct. 2021, pp. 121-136.

[3] F. Abri, S. Siami-Namini, M. A. Khanghah, F. M. Soltani and A. S. Namin, "Can Machine/Deep Learning Classifiers Detect Zero-Day Malware with

High Accuracy?," *2019 IEEE International Conference on Big Data (Big Data*), Los Angeles, CA, USA, 2019, pp. 3252-3259, https://doi: 10.1109/BigData47090.2019.9006514.

[4] Y. Jung, "A review of privacy-preserving human and human activity recognition," Int. J. Smart Sens. Intell. Syst., vol. 13, no. 1, pp. 1-13, 2020.

[5] A. Etuk, J. Anyadighibe, E. James, and R. Mbaka, "Service quality and passengers' loyalty of public transportation companies," *British Journal of Management and Marketing Studies*, vol. 4, no. 4, pp. 82-98, 2012.

[6] Y. Guo, "A Survey of Machine Learning-Based Zero-Day Attack Detection: Challenges and Future Directions", Comput Commun. 198(C), 175–185, 2023, https://doi:10.1016/j.comcom.2022.11.001.

[7] M. Sarhan, S. Layeghy, M.R. Gallagher, M. Portmann, "From zero-shot machine learning to zero-day attack detection." *Int. J. Inf. Secur.* 22, pp. 947–959, 2023. https://doi.org/10.1007/s10207-023-00676-0 [8].

[8] A. Ekong, B. Ekong, A. Edet, "Supervised machine learning model for effective classification of patients with covid-19 symptoms based on bayesian belief network", *Researchers Journal of Science and Technology*, vol2: pp. 27 – 33, 2022.

[9] V. C. Victor, C. Ugwu, and O. Laeticia Onyejegbu, "Enhanced Classification Model for Likelihood of Zero-Day Attack Detection and Estimation," *European Journal of Electrical Engineering & Computer Science*, vol. 5, no. 4, 2021.

[10] S. Ali, S. U. Rehman, A. Imran, G. Adeem, Z. Iqbal, and K.-I. Kim, "Comparative Evaluation of AI-Based Techniques for Zero-Day Attacks Detection," *Electronics*, vol. 11, no. 23, 2022, Art. no. 3934, https://doi: 10.3390/electronics11233934.

[11] V. T. Victor, C. U. Chidiebere, and O. Laticia, "Comparative Analysis of Dimensionality Reduction Techniques on Datasets for Zero-Day Attack Vulnerability," *Journal of Software Engineering and Simulation*, vol. 7, no. 8, pp. 48-56, 2021.

[12] J.-Y. Kim, S.-J. Bu, and S.-B. Cho, "Zero-day malware detection using transferred generative adversarial networks based on deep autoencoders," *Information Sciences*, vol. 460, pp. 83-102, 2018.

[13] A. Ekong, E. Udo, O. Ekong, and S. Inyang, "Machine Learning based Model for the Prediction of Fasting Blood Sugar Level towards Cardiovascular Disease Control for the Enhancement of Public Health," *International Journal of Computer Applications*, vol. 975, pp. 8887.

[14] D. Feaster (2018). Malware Detection. Kaggle. [Online]. Available: https://kaggle.com/competitions/malware-detection.

[15] Verywell Mind. "The Algorithm Problem Solving Approach in PsychologyVerywellMind."[Online].Available: https://www.verywellmind.com/what-is-an-algorithm-2794807. Accessed on: Aug. 24, 2023.

[16] S. S. Raut and S. S. Kulkarni, "Random Forest Modeling for Network Intrusion Detection System," in Procedia Computer Science, vol. 89, pp. 797-803, 2016, doi: 10.1016/j.procs.2016.06.103.