Towards Optimization of Malware Detection using Chi-square Feature Selection on Ensemble Classifiers

Fadare Oluwaseun Gbenga, Adetunmbi Adebayo Olusola, Oyinloye Oghenerukevwe Eloho, Mogaji Stephen Alaba

***Abstract:** The multiplication of malware variations is probably the greatest problem in PC security and the protection of information in form of source code against unauthorized access is a central issue in computer security. In recent times, machine learning has been extensively researched for malware detection and ensemble technique has been established to be highly effective in terms of detection accuracy. This paper proposes a framework that combines combining the exploit of both Chi-square as the feature selection method and eight ensemble learning classifiers on five base learners- K-Nearest Neighbors, Naïve Bayes, Support Vector Machine, Decision Trees, and Logistic Regression. K-Nearest Neighbors returns the highest accuracy of 95.37%, 87.89% on chi-square, and without feature selection respectively. Extreme Gradient Boosting Classifier ensemble accuracy is the highest with 97.407%, 91.72% with Chi-square as feature selection, and ensemble methods without feature selection respectively. Extreme Gradient Boosting Classifier and Random Forest are leading in the seven evaluative measures of chi-square as a feature selection method and ensemble methods without feature selection respectively. The study results show that the tree-based ensemble model is compelling for malware classification.*

***Keywords:** Chi-square, Extreme Gradient Boosting Classifier, K-Nearest Neighbors, Random forest.*

I. INTRODUCTION

The protection of information in form of source code against unauthorized access is a central issue in computer security. In this period of software globalization, the requirement for making sure about software is abundantly looked to guarantee its smooth working for nonstop accessibility of administrations to the whole clients. As different PCs are associated with an overall organization, the software is an objective of copyright privateers, assailants, or even fear-based oppressors, accordingly, software securities become an appropriate issue for software clients and engineers.

Manuscript received on March 25, 2021.
Revised Manuscript received on April 29, 2021.
Manuscript published on April 30, 2021.
* Corresponding Author

Fadare Oluwaseun Gbenga*, Dept. Of Computer Science, Joseph Ayo Babalola, Ikeji-Arakeji. Osun-State. Nigeria.

Prof. Adetunmbi Adebayo Olusola, Dept. of Computer Science, Federal University of Technology, Akure. Ondo-State. Nigeria.

Dr. (Mrs) Oyinloye Oghenerukevwe Eloho, Dept. of Computer Science, Ekiti State University, Ado-Ekiti. Ekiti-State. Nigeria.

Dr. Mogaji Stephen Alaba, School of Computing, Federal University of Technology, Akure. Nigria.

The malware identifies with pernicious software culprits dispatch to contaminate singular PCs or a whole association's organization. It uses target framework weaknesses, for example, a bug in real software(e.g., a program or web application module) that can be captured. Malware event happens in associations at regular intervals, assaults numerous segments with disturbing misfortunes to protected innovation, bargained client records, and even obliteration of information [1]. Malware detection refers to the cycle of detecting the presence of malware on a framework or of recognizing whether a particular program is noxious or considerate. Their spread is quick, [2] with the capacity to taint upwards of 359,000 PCs in less than 14 hours, or much quicker. PC malware subsequently presents one of a kind difficulties to security specialists. With the persistent improvement of data innovation, cybercrime is a genuine danger to the monetary, military and other significant regions of different nations. Malware is one of the significant components that sabotage Internet security. Malware has developed quickly in both amount and classifications contrasted and foretime. New malware, particularly Advanced Persistent Threat (APT), is increasingly more hard to be identified by current abandonment innovations. The ensemble of malware is vital to distinguish new malware. There are different improvements that happen in the arrangement when applied various feature selection approaches and machine learning techniques. The feature selection technique assumes a fundamental function in classification as it limits handling time, improves exactness, diminishes information dimensionality, and eliminates insignificant highlights preceding classification [3]. Feature selection determination is the way toward choosing a subset of important highlights for use in model development. To improve the performance, ensemble strategies are profoundly powerful and give better consequences of accuracy. Machine learning methods acquire greater headway in classification, and ensemble learning gives ideal arrangements by consolidating base models of AI [4]. This malignant software taints a great many PCs consistently, and the Microsoft Windows working framework (OS) remains the most influenced, as it is the most utilized OS around the world. In this paper, we propose a structure that joining the endeavor of both chi-square as a feature selection method and eight ensemble learning classifiers on five base learners - KNN, Naive Bayes, SVM, Decision Trees, and Logistic Regression. This paper is sorted out into five areas. The following segment talks about related work.

The third area examines the methodology while the fourth segment talks about the experimental setup and result discussion. The fifth segment talks about the conclusion.

II. RELATED WORK

Over the latest twenty years, researchers have made a few endeavors in identifying malware utilizing machine learning approaches. A couple of these endeavors are examined beneath: Alazab [5] announced that numerous Internet of Things (IoT) administrations are presently followed and controlled through cell phones, making them helpless against protection assaults and misuse by different pernicious applications. The author proposed a model for distinguishing malicious applications dependent on genuine world datasets. Two-component choice techniques—Chi-Square and ANOVA—were inspected related to ten administered machine-learning algorithms. Chi-Square was found to have a higher discovery accuracy when contrasted with ANOVA. The proposed framework accomplished a detection accuracy of 98.1% with a grouping season of 1.22 s. Seoungyul et. al. [6] thought progresses in machine learning algorithms have improved he performance of malware identification frameworks for the most recent decade. In any case, there are still a few difficulties, for example, handling a lot of malware, learning high-dimensional vectors, high stockpiling utilization, and low adaptability in learning. This paper proposed low-measurement yet powerful highlights for a malware identification framework and investigates them with tree base gathering models. Eslam and Ivan [7] contended that current business antivirus detection engines actually depend on signature-based strategies. The authors proposed model utilize just the base minimal set of features to make a grouping design that can distinguish malware. The model could anticipate concealed malware with an exactness pace of 0.998 and with a false positive pace of 0.002. Ochieng et. al. [8] thought that advanced PC worm jumble the code to make it hard to recognize. The examination strayed from existing recognition approaches by utilizing dim space network traffic ascribed to a real worm assault to prepare and approve the machine learning algorithms. It was likewise gotten that the different ensemble performs similarly well. HarshaLatha and Mohanasundaram [9] proposed another hybrid methodology that joins include feature selection with ensemble learning methods to improve accuracy for high dimensionality information. They utilized various feature selection methods: Percentile, Extra Trees Classifier, and KBest include choice techniques to choose the best highlights (dimensionality decrease) and with four ensemble classifiers. Ada Boost, Gradient Boosting, Random Forest, and Bagging were utilized for classification. Among all the outcomes, the Hybrid model with a random forest classifier gives a better outcome of 91.50 % accuracy.

In completely related works, there is nobody that considering a solitary feature determination strategy as against numerous ensemble classifiers. This paper proposes a system joining the endeavor of Chi-square as feature selections and eight ensemble methods on five base students KNN, Naive Bayes, SVM, Decision Trees, and Logistic Regression.

III. METHODOLOGY

The proposed architecture malware detection portrayed in Fig. 1 includes four stages. Data acquisition and preprocessing, feature selection, model building, and assessments. Dataset was extricated utilizing python programming language and it was pre-prepared. Feature selection strategy was applied to the dataset. Dataset was normalized with standard scaler utilizing the python programming language. Dataset was part into a proportion of 70% for the training set and 30% for the testing set. The training set is a dataset that machine learning algorithms and ensemble algorithms must act upon. The dataset we use to test the accuracy of our model is called the testing dataset. The dataset we use to test the accuracy of our model is known as the testing dataset. Base classifiers (KNN, Naive Bayes, SVM, Decision Trees and Logistic Regression). The training set and testing data were applied on the ensembles (Bagging, AdaBoost, Gradient Boosting, Extreme Gradient Boosting Classifier, Light Gradient Boosting Classifier, Voting, Extra tree, and random forest). The feature selection method chosen for this study is chi-square. We train and test the dataset with five classification algorithms. Training data are used to fit and tweak the models. Then we train and test the dataset with eight ensemble classifiers. The model was assessed with the seven evaluative measures. Fig. 2 uncovers the proposed malware identification algorithms.

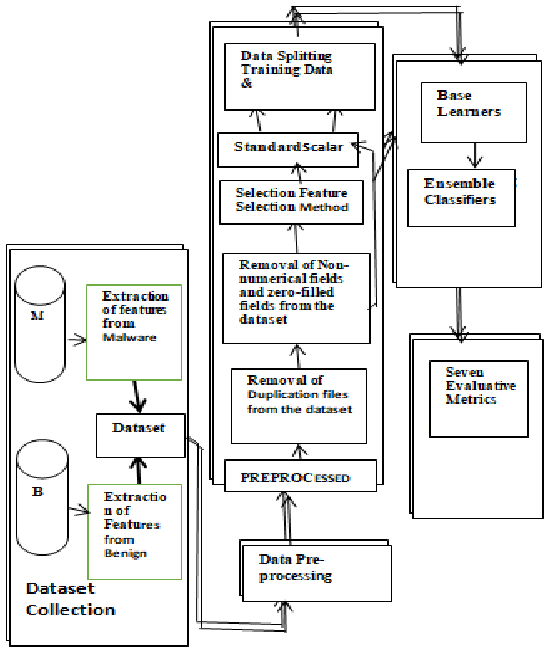
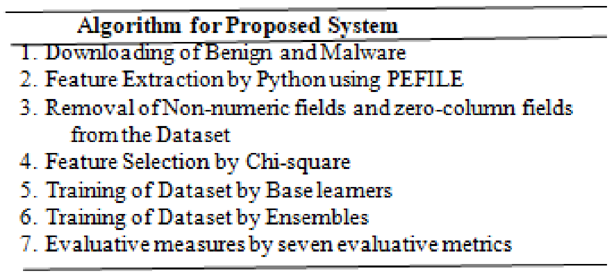

Data Splitting Training Data &
Base Learners
Standard Scalar
Selection Feature Selection Method
Ensemble Classifiers
Removal of Non-numerical fields and zero-filled fields from the dataset
M
Extraction of features from Malware
Seven Evaluative Metrics
Removal of Duplication files from the dataset
Dataset
PREPROCESSED
B
Extraction of Features from Benign
Data Pre-processing
Dataset Collection

Fig. (1). An overview of proposed malware detection model


Algorithm for Proposed System

1. Downloading of Benign and Malware
2. Feature Extraction by Python using PEFILE
3. Removal of Non-numeric fields and zero-column fields from the Dataset
4. Feature Selection by Chi-square
5. Training of Dataset by Base learners
6. Training of Dataset by Ensembles
7. Evaluative measures by seven evaluative metrics

Fig. (2). Algorithm for the proposed Malware Detection System

A. Data Description

Portable Executables of 9,428 of 49.3 G of kind files are gotten from clean window applications 32 bits and 64 bits Windows 7 framework, Windows XP, and downloaded from different considerate sites archives: Ninite [10], Downloads [11], and Softpedia [12]. Totalvirus [13] is utilized to examine all the downloaded documents to find out if the records are really benevolent or malware. Totalvirus [13] contains very nearly 80 AV engines, just downloaded documents with zero recognized rate from all the 80 AV engines were chosen. These are benign files. 89 G of 14,247 samples of malware were downloaded from Virushare [14] and Virussign [14], online repositories. Table 1 spells the dissemination of the ten malware types in the dataset. The mix of malware and benign brought about an aggregate of 23,675 instances utilized for test study. The inescapable of Windows bears it an enrapturing climate for malware makers to compose malicious codes. In this paper, we focus on the convenient executable configuration for 64-bit Windows working frameworks. In this paper, we just consider non-packed programs. Packing is a procedure that is utilized lawfully by programming designers to build their programs from figuring out and malware writers use it to hide the vindictive program from being identified by AV engines [16]. The binary portable compact executables are the main executables that were isolated for our data. (83) features are extracted using standardized PE File format from the dataset by using a python program. Thirteen features were dropped and they are target column, a column named "Name of file", non-numeric data type, and zero-column fields staying seventy (70) features.

Table 1. Malware Data Type

No	Malware Type	Counts	No	Malware Type	Counts
1	Trojan	2493	6	Backdoor	983
2	Trojan-Dropper	1353	7	Ransomware	1350
3	Trojan-Spy	755	8	Spyware	996
4	Virus	2165	9	Email-worm	1089
5	Worm	1601	10	Exploit	1462
TOTAL = 14247					

B. Feature Selection

Feature selection methods are explicitly helpful for some reasons like dimensionality decrease, improve exactness, eliminate unimportant features and lessen the computational or handling time [3], [17].

C. Chi-Square

Chi-square is another filter feature choice techniques that evaluates association of 2 categorical variables. Thus, the independent variable desires numeric variables. The Chi-square datum is obtained as follows[18]:

$$x^2 = \sum_{i=1}^{m} \sum_{i=1}^{k} \left(\frac{A_{ij} - E_{ij}}{E_{ij}} \right) \qquad (1)$$

where m denotes the number of intervals and k denotes the number of classes, A_{ij} represents the number of samples in the i^{th} interval *jth* class, R_i represents the number of samples in the i^{th} interval, C_j represents the number of samples in the jth class, *N* represents the total number of samples, and is the predicted frequency of $A_{ij}\left(E_{ij} = R_i * C_j/N\right)$. In general, the larger the measured chi-squared value, the more significant the feature is

Table 2. Feature Selection by Chi-Square

No	Data fields	Attributes	Scores
1	11	ImageBase	2.122929e+12
2	3	CheckSum	2.146061e+10
3	46	SizeOfInitializedData	1.476041e+10
4	45	SizeOfImage	1.432618e+10
5	29	ResSize	1.365021e+10
6	33	SectionMaxVirtualSize	1.245473e+10
7	39	SectionMinRawSize	1.245473e+10
8	32	SectionMaxRawSize	9.231928e+09
9	7	ExportRVA	4.557391e+09
10	10	IATRVA	2.704181e+09
11	36	SectionMeanVirtualSize	2.080219e+09
12	35	SectionMeanRawSize	1.561290e+09
13	0	AddressOfEntryPoint	1.545557e+09
14	14	LoaderFlags	9.914608e+08

D. Base Learners

Here, various Base Learners algorithms inspected in this experiment are quickly portrayed as follows:

E. Naïve Bayes

Schultz et. al.[19] concurred that Naïve Bayes is a probabilistic technique. Given an obscure testing example, it utilizes (2) to register back the likelihood of each class and afterward the class with the most elevated worth is its forecast

$$C = arg_{C_i} maxP(C_i) \prod_J P(F_J/C_i) \qquad (2)$$

Where P(Ci) is the likelihood of class i, P(Fj|Ci) is the contingent likelihood of each component esteem given the class i

F. Support Vector Machine

Konstantinou and Wolthusen [20] believed that the Support Vector Machine depends on the auxiliary danger minimization standard from the factual learning hypothesis. It is especially appropriate for taking care for solving binary classification problems. SVM can distinguish an ideal isolating hyper-plane between two classes in a high measurement include space. The ideal hyper-plane is dependent upon the limitation as:

$$min = \frac{1}{2} \parallel w \parallel^2 + C \sum_{i=1}^{l} \xi_i \qquad (3)$$

s.t $y_i(w.x_i + b) \geq 1 - \xi_i$, $\xi_i \geq 0$ $\quad i = 1, \ldots . l$

where w is ordinary to the hyper-plane, is a mistake for the i-th occasion. C is a boundary to speak to the tradeoff between augmenting the edge and minimizing the training error.

G. K-Nearest Neighbor (KNN)

K-Nearest Neighbor (KNN) is the most clear least simplest machine learning algorithm that is used for both classification and relapse models. At whatever point the model is attempted with testing data it finds the partition of that point from one another point in the training data [21]. By then, it finds the nearest k people for that point. The use of KNN should be conceivable by following a couple of stages which are given underneath:
Weight the data.
Instate the evaluation of k.
For getting the anticipated class, highlight from 1 to mean the amount of planning data centers [21]. KNN uses neighborhood gathering as the gauge assessments of the new request model. It glances through the model space for the k getting ready tuples that are closest to the dark tuple. Closeness is portrayed by a detachment metric, for instance, euclidean division. The Euclidean detachment between two concentrations or tuples state,

$$X_1 = (x_{11}, x_{12} \dots \dots) \; and \; x_2 = (x_{21}, x_{22}, \dots, x_{2n}) \qquad (4)$$

is $dist(X_1, X_2) = \sqrt{\sum_{i-1}^{n} (x_{1i} - x_{2i})^2}$

H. Logistic regression algorithm

A logistic regression algorithm is used to predict discrete or straight out characteristics. It is basically a characterization calculation that is used in cases like deception distinguishing proof, email spam acknowledgment among others, where we have to make decisions among yes and no. It predicts the likelihood of the event of an occasion by fitting information into the justification work [21]. Regularly, a logistic regression model determines the class enlistment probability for one of the two orders in the educational assortment [22]. Regardless, the Logistic curve is unquestionably not a straight twist like an immediate backslide. It is known as the sigmoid curve and here probability.

$$\boldsymbol{p = 1/1 - e^{\wedge} - z} \qquad 5$$

where z= mx+c. This condition of likelihood guarantees that the indicator will be somewhere in the range of 0 and 1.

I. C4.5 Decision Tree

C4.5 calculation is started from ID3 calculation, an exceptionally straightforward choice tree calculation, introduced by [23]. This calculation goes the through decision tree, visits every hub, and select the ideal split. It is accomplished by utilizing the gain ratio, spoken to by following recipe [23].

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo(A)} \qquad 6$$

Gain or information gain is a quality determination measure utilized in ID3 approach. In information gain, a property with the most elevated data gain is picked as a parting trait for the hub N. Thus, this trait limits the data expected to characterize tuples D in a parcel and returns the least "contamination" in these segments. Information gain is a distinction in entropy from before to after the set D is parceled on characteristic A. Additionally, it checks how much vulnerability in D is decreased after it is parceled on characteristic A. The vulnerability in the data set D is estimated by entropy determined as:

$$\boldsymbol{Entropy(D) = -\sum_{x \in X} p(x) log_2 p(x)} \qquad 7$$

where X is the arrangement of classes in D and p(x) is the extent of the number of components in class x to the number of components in set D. At the point when entropy is 0, the informational index is totally arranged [24]. SplitInfo is the term that depicts how similarly the property parts the information and is determined the equation [24].

$$\textbf{SplitInfo}(\mathbf{A}) = -\sum_{j=i}^{n} \frac{|D_j|}{|D|} \times log_2 \left(\frac{|D_j|}{|D|}\right) \qquad 8$$

The term $\frac{|D_j|}{|D|}$ addresses the weight of jth section.

J. Ensembles

Ensemble learning is a powerful part of machine learning that is utilized to improve the accuracy and performance of conventional AI classifiers [25]. It works by making a center gathering of base learners and joining their yields for ultimate choice creation. Ensemble learning exploits correlative data of various classifiers to improve the accuracy and performance of the choice. The ensemble techniques researched included Gradient Boosting Classifier, AdaBoost, Bagging, Extreme gradient Boosting Classifier, Light Gradient Boosting Classifier, Random Forests, and Extra Trees Classifier. Furthermore, their concise portrayals are given beneath

K. Adaptive boosting

Adaptive boosting (AdaBoost) at first relegates equivalent loads to each training perception. It employments various weak models and gives out higher models to those observations for which misclassification was viewed. As it utilizes numerous powerless models, joining the consequences of the choice limits accomplished during various cycles, the accuracy of the misclassified perceptions is improved, and thus the accuracy of overall iterations are also improved [26].
The weak models are assessed utilizing the mistake rate as given in condition (9)

$$\boldsymbol{\varepsilon_t = Pr_{\sim D_t} \left[h_t(X_i) \neq Yi \right] = \sum_{i: h_{t(X_i)} \neq Yi} D_t} \qquad 9$$

where ε_t is the weighted blunder gauge, $Pr_{\sim D_t}$ is the likelihood of the irregular model i to the circulation Dt, h_t are the speculations of the weak learners, X_i is the preparation perception, Yi is the objective variable, t is the emphasis number. The prediction error is one of the classification is wrong and 0 if the classification is correct.

L. Extra Trees Classifier

Extra Trees Classifier is a tree-based outfit classifier. It joins the gathering of decision trees known as "forest" and produces the arrangement yield Chen and Guestrin [27]. Every decision tree is shaped by the preparation test. The Extra-Tree procedure (speaking to incredibly randomized trees) was proposed with the essential objective of further randomizing trees working concerning mathematical data highlights, where the choice of the ideal cut-point is responsible for a tremendous degree of the vacillation of the instigated tree.

M. Random Forest

Random Forest uses an augmentation to the Bagging approach [28]. In Bagging, every classifier is constructed separately by working with a bootstrap test of the information. In a normal choice tree classifier, a choice at a hub split is made dependent on all the component ascribes. In any case, in Random Forest, the best boundary at every hub in a choice tree is produced using a randomly selected number of features. This arbitrary determination of selection of features encourages random forest models to not just scale well when there exist numerous features per include vector, yet in addition to diminishing the reliance (relationship) between's the component ascribes. Random forest utilizes the data substance of a hub as the parting basis. The information content is characterized as:

$$I(N) = |S|H(S) - |S_L|\, H(S_L) - |S_R|H(S_R) \qquad 10$$

Where |S| = input size,
$|S_{I,\ R}|$ = Size of left. Right Subclasses of S.

$$H(S) = Shannon\ entropy\ of\ S = \sum_{i=\pm 1}^{m} (p_i) log_2 (p_2) \quad 11$$

N. Gradient Boosting

Gradient Boosting or GBM is another ensemble machine learning algorithm that works for both relapse and classification issues [29]. GBM utilizes the boosting strategy, consolidating various powerless weak learners to form a powerful learner. It is a voracious algorithm and can overfit a preparing dataset rapidly. It can benefit from regularization strategies that punish different pieces of the algorithm and by and large improve the performance of the algorithm by reducing overfitting.

O. Extreme gradient boosting

Extreme gradient boosting is the improvement of gradient boosting which depends on the boosting algorithm and considered as an ensemble classifier. It enhanced the gradient boosting execution to work quicker and added regularization boundaries to abstain from overfitting. This investigation sets the learning rate for each boosting round to 0.01 and set the number of supported trees to 200. XGBoost has ended up being an exceptionally powerful ML algorithm, broadly utilized in machine learning rivalries. XGBoost has a high prescient force and is right around multiple times faster than the other gradient boosting methods [27].

P. LightGBM Classifier

LightGBM Classifier Gradient boosting is one of the machine learning algorithms utilized for classification and relapse. It consolidates models from various algorithms to deliver a new iterative one. LightGBM Gradient boosting is one of the most fiercely utilized machine learning algorithms because of its exactness and effectiveness [30], [31]. LightGBM is a gradient boosting system that utilizations tree-based calculations and follows leaf-wise methodology while different calculations work in a level-wise methodology design.

Q. Voting

Expecting that a classifier ensemble comprises of L base Classifiers in the set $\boldsymbol{D = \{D_1, \dots\dots, D_L\}}$, and any item $\boldsymbol{x \in R^n}$ is allotted to one of the $\boldsymbol{c}$ potential classes $\boldsymbol{\Omega = \{\omega_{1,}\ \dots\dots.., \omega_c\}}$. For x to be arranged, L classifiers a matrix $\boldsymbol{M = [m_{i,j}],\ i = 1, \dots\dots, L,\ j = 1, \dots\dots., c}$,. Assume $\boldsymbol{m_{i,j}} \in \{\boldsymbol{0,1}\}$, where $\boldsymbol{m_{i,j} = 1}$ if $\boldsymbol{D_i}$ predicts x in $\boldsymbol{\omega_j}$ class, and $\boldsymbol{m_{i,j}}$ =0, in any case x is relegated to if

$$\sum_{i=1}^{L} m_{i,k} = \max_{j=1}^{c} \sum_{i=1}^{L} m_{i,j} \qquad 12$$

This standard is called the majority voting rule [32]. Assume $\boldsymbol{m_{i,j}} \in \{\boldsymbol{0,1}\}\ \boldsymbol{where\ m_{i,j}}$ is the level of help that classifier $\boldsymbol{D_i}$ provides for the theory that x originates from class $\boldsymbol{\omega_j}$, meant as $\boldsymbol{m_{i,j} = P_{Di}(\omega_j|x)}$. X is alloted to $\boldsymbol{\omega_k}$ if

$$\frac{1}{L}\sum_{i=1}^{L} m_{i,k} = \max_{j=1}^{c} \frac{1}{L}\sum_{i=1}^{L} m_{i,j} \qquad 13$$

R. Bagging

Bagging is acronyms for Bootstrap Aggregation. It is a straightforward and exceptionally incredible ensemble technique for improving tender assessment or identification grouping. [28] inspired bagging as a variance decrease strategy for a given base methodology, for example, choice trees or techniques that do variable determination and fitting in a one-dimensional model. It is an approach to diminish the variance in the expectation by creating extra information for training from dataset utilizing mixes with reiterations to deliver multi-sets of the first data [33]. By then, the chance of ensemble methods is to try diminishing tendency and moreover variance of such weak models by joining a couple of them together to make a strong learner (or ensemble model) that achieves better displays [34].

Various machine learning ensembles were explored and their detection capabilities were investigated. The ensemble methods investigated included Gradient Boosting Classifier, AdaBoost, Bagging, Extreme gradient Boosting Classifier, Light Gradient Boosting Classifier, Random Forests, Voting, and Extra Trees Classifier.

IV. PERFORMANCE METRICS

For assessment reason, we utilized the Overall Accuracy (OA), False Positive, False Negative, True Positive, True Negative, Recall, Precision, F1-Score, False Positive Rate, ROC, Cohen Kappa, and AUC. False Negatives (FN): the number of malicious samples classified as benign. True Negatives (TN):

the number of benign samples classified as benign. False Positive (F.P) implies wrongly classifier favorable as malware. True Negative (T.N) means correctly classify benign as benign. The recall is the capacity of a calculation to locate every single positive example. A recall is equivalent to a True Positive rate. Precision is the proportion of accurately anticipated positive to all out the anticipated positive example. F1-Score is the weighted normal of Precision and Recall. As the accuracy and recall in characterization measure is a couple of contradictory measures, the utilization of F1-Score can adequately adjust the precision and review, and the more like 1 of F1-Score mathematical worth methods better classifier execution. False Positive Rate is otherwise known as the probability of false alarm. Accuracy score= (TP+TN)/(TP+TN+FP+FN). Precision=TP/(TP+FP), Recall=TP/(TP+FN). The ROC (AUC), Area of a classifier is the likelihood of the classifier positioning an arbitrarily picked positive occasion higher than a haphazardly picked negative occurrence. The model is assessed through a few presentation assessment measures, specifically exactness, recall, accuracy f1-score, true positive rate, roc., cohen kappa, and AUC.

V. EXPERIMENTAL SETUP AND RESULT DISCUSSION

The dataset was split in the ratio of 70:30 for the training and testing. Seventy percent (70%) of each member type of malware consulting 9,973 of malware and 6,600 of benign were used for the training while testing data comprising of 4,274 of malware and 2,828 of benign were used for the experimental setup. Benign was set to 1 and malware was set to 0, so experimental was based on a binary-class experiment. Feature selection on the training set is chi-square. It selects 14 best features among 70 features based on set threshold. Table 2 shows the feature selection by chi-square. For without feature selection, we used the resulted seventy (70) features extracted from the dataset. Table 3 shows the results of base learners with feature selection techniques by chi-square and without feature selection.

Table 3. Confusion Matrix and Accuracy for the base learners with and without feature selection

Base Learners	Feature Selections	TP	TN	FP	FN	Accuracy
Logistic Regression	Chi-square	3692	2454	582	374	86.54
	Without Feature Selection	3438	2260	836	568	80.23
Decision Tree	Chi-square	4050	2722	224	106	95.35
	Without Feature Selection	3754	2463	520	365	87.54
Support Vector Machine	Chi-square	3695	2464	580	364	86.71
	Without Feature Selection	3438	2260	836	568	80.23
Naive Bayes	Chi-square	3524	2378	750	450	83.81
	Without Feature Selection	3252	2162	1022	666	76.23
K-nearest neighbors	Chi-square	4051	2722	223	106	95.37
	Without Feature Selection	3768	2474	506	354	87.89

Regarding accuracy, Table 3 shows that the KNN classifier returned the most noteworthy accuracy of 95.37% with chi-square as a feature selection technique. KNN and Decision Tree had the highest among the base learners. Different classifiers like logistic regression, support vector machine, and naive bayes additionally indicated significant-high accuracy outcomes. KNN classifier demonstrated transitional outcomes contrasting with the previously mentioned classifiers. Naive Bayes had the lowest of 83.81% accuracy with chi-square as a feature selection technique. Also, from table 3, experiment with the dataset without feature selection shows that KNN had the highest of 87.89% in terms of accuracy while Naïve Bayes had the lowest of 76.23% accuracy. In Logistic Regression, chi-square returned 86.54% of accuracy which is higher than of without feature selection which returned 80.23% of accuracy. In Decision Tree Classifier, chi-square returned the higher of 95.35% while without feature selection returned 87.54% of accuracy. In Support Vector Machine, chi-square returned 86.70% of accuracy while without feature selection returned the lower of 80.23% of accuracy. In Naïve Bayes, chi-square returned 83.81% of accuracy while that of without feature selection returned lower of 76.23% of accuracy. In K-nearest neighbors, chi-square returned 95.37% of accuracy while the without feature selection returned 87.89% of accuracy. This shows that the feature selection technique (chi-square) has a greater influence on detecting the accuracy of the classification.

Table 4. Confusion Matrix and Accuracy for the Ensemble methods with and without feature selection

Ensemble Methods	Feature selection Techniques	TP	TN	FP	FN	Accuracy
Bagging	Chi-square	4132	2781	142	47	97.34
	Without Feature Selection	3869	2539	405	289	90.23
Ada Boosting	Chi-square	4118	2765	156	63	96.92
	Without Feature Selection	3824	2490	450	338	88.90
Gradient Boosting	Chi-square	4118	2765	156	63	96.92
	Without Feature Selection	3835	2498	439	330	89.17
Extreme Gradient Boosting Classifier	Chi-square	4135	2782	139	46	97.40
	Without Feature Selection	3926	2588	348	240	91.72
Light Gradient Boosting Classifier	Chi-square	4133	2780	141	47	97.35
	Without Feature Selection	3879	2554	395	274	90.58
Majority Voting	Chi-square	3858	2519	416	309	89.79
	Without Feature Selection	3681	2447	593	381	86.29
Random Forest	Chi-square	4134	2781	140	47	97.37
	Without Feature Selection	3895	2566	379	262	90.97
Extra Tree	Chi-square	4128	2779	146	49	97.25
	Without Feature Selection	3867	2538	407	290	90.19

Table 4 is the table for confusion matrix, accuracy for the ensemble methods with and without feature selection (chi-square). The extreme gradient boosting classifier had the highest accuracy of 97.40% of all ensemble approaches

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication
© Copyright: All rights reserved.

with feature determination strategy (chi-square) while the ensemble method without the feature selection method had the most noteworthy accuracy of 91.72% of all ensemble approaches. Followed by Random Forest which had the accuracy of 97.37% of all ensembles with feature determination strategy (chi-square) while the light gradient boosting classifier had the accuracy of 97.35% of chi-square as a feature selection technique. Next is the bagging which had an accuracy of 97.34% of detection rate of all ensembles with feature selection technique (chi-square) while voting had the lowest detection rate of 89.79% of all ensemble approaches with feature selection technique. Random Forest had an accuracy of 90.97% without the feature selection technique. Followed by light gradient boosting classifier which had 90.58 accuracy while voting had the lowest of 86.29 accuracy without feature selection technique. The feature selection technique (chi-square) of all the ensembles is higher than the ensemble without feature selection.

Table 5. Seven evaluative measures of all ensembles for Feature Importance by Chi-Square

Feature Importance by Chi-square	Precision	F1-Score	False Positive rate	ROC	Cohen Kappa	AUC-
Bagging	0.96678	0.97764	0.04858	0.98870	0.86228	0.95159
Ada Boosting	0.96350	0.97410	0.05341	0.97689	0.84013	0.92496
Gradient Boosting	0.96350	0.97410	0.05341	0.97689	0.84013	0.92496
Extreme Gradient Boosting	0.96748	0.97812	0.04759	0.98792	0.85647	0.94978
Light Gradient Boosting	0.96701	0.97776	0.04827	0.98690	0.85527	0.94710
Majority Voting	0.90267	0.91411	0.14174	0.97345	0.83236	0.91289
Random Forest	0.96724	0.97788	0.04793	0.98973	0.86281	0.95474
Extra Tree	0.96584	0.97693	0.04991	0.98784	0.85677	0.94797

Table 5 shows the seven parametric measures of the ensembles with the feature selection technique of random forest. Extreme Gradient Boosting recorded the highest of all the evaluative traits, 0.98900 in the recall, 0.96748 in precision, 0.97812 in f1-score, false positive-rate of 0.04827, while random forest recorded roc of 0.98973, cohen kappa of 0.86281, and AUC of 0.95474. Majority Voting recorded the lowest of all the evaluative traits with the recall of 0.92585, the precision of 0.90267, f1-score of 0.91411, false-positive rate of 0.14174, roc of 0.97345, cohen kappa of 0.83236, and AUC of 0.91289.

Table 6. Seven evaluative measures of all ensembles for Without Feature Selection Method

Without selection feature	**Preci-sion**	**F1_Score**	**False positive rate**	**ROC**	**Cohen Kappa**	**AUC-**
Bagging	**0.9052**	**0.9176**	**0.1376**	**0.8392**	**0.7302**	**0.8308**
Ada Boosting	**0.8947**	**0.9065**	**0.1531**	**0.8272**	**0.7242**	**0.8180**
Gradient Boosting	**0.8973**	**0.9088**	**0.1495**	**0.8311**	**0.7107**	**0.8242**
Extreme Gradient Boosting	**0.9186**	**0.9303**	**0.1185**	**0.8381**	**0.7281**	**0.8302**
Light Gradient Boosting	**0.9076**	**0.9205**	**0.1339**	**0.8321**	**0.7267**	**0.8272**
Majority Voting	**0.8613**	**0.8832**	**0.1951**	**0.8198**	**0.7089**	**0.8091**
Random Forest	**0.9113**	**0.9240**	**0.1287**	**0.8418**	**0.7312**	**0.8314**
Extra Tree	**0.9048**	**0.9173**	**0.1382**	**0.8371**	**0.7298**	**0.8293**

Table 6 shows the seven parametric proportions of the ensembles without the feature determination procedure strategy. Extreme Gradient Boosting recorded the highest in four evaluative traits out of seven, recall of 0.9424, the precision of 0.9186, f1-score of 0.9303, and false positive rate of 0.1185 while the rest of the traits went to the random forest. Majority Voting returning the lowest with the recall of 0.9062, the precision of 0.8613, f1-score of 0.8832, false-positive rate of 0.195, roc of 0.8198, Cohen Kappa of 0.7089, and AUC of 0.8091.

Table 7 shows the seven evaluative measures of the ensembles with their feature selection techniques. In the Bagging ensemble, seven evaluative measures of chi-square are greater that the ensemble without feature selection. In Ada Boosting, seven evaluative measures of chi-square are greater that the ensemble without feature selection. In Gradient Boosting, seven evaluative measures of chi-square are greater that the ensemble without feature selection. In all the ensembles, even evaluative measures of chi-square are greater that the ensemble without feature selection. From Table 5 and Table 6, Extreme Gradient Boosting Classifier and Random forest are paramount predominant which returned the highest. XGBoost is an effective AI model, which has the upsides of sparing assets, less preparing time and high exactness, and comprehends numerous difficulties. XGBoost is a group based estimation which has been helping a huge amount of data scientists to win Kaggle contentions [35]. It is an extremely fast algorithm and supports the parallel building of the forest. It works on the principle of Gradient Boosting.

Random forest is a fruitful ensemble classifier dependent on the bootstrap accumulation algorithm and choice tree. The investigation of [36],[37] demonstrates that random forest can document the best outcome contrasting with different classifiers. LightGBM Classifier Gradient boosting is one of the machine learning algorithms used for classification and regression. It joins models from various algorithms to deliver a new iterative one. Bagging is the utilization of the Bootstrap technique to a high-variance machine learning algorithm, ordinarily decision trees. The predisposition change compromise is a test we as a whole face while training machine learning algorithms. Bagging is an amazing ensemble technique that assists with the diminishing change, and by expansion, forestall overfitting.

VI. CONCLUSION

Chi-square is used as feature selection technique and trained it on the five base learners and eight ensembles. Table 3 shows that the KNN classifier returned the highest accuracy of 95.37% with chi-square as feature selection technique while KNN classifier returned the highest accuracy of 87.89% without feature selection technique.

Table 4 shows Extreme Gradient Boosting Classifier had the highest accuracy of 97.40% of all ensemble approaches with feature selection technique (chi-square) while ensemble methods without the feature selection had the accuracy of 91.72% of all ensemble approaches. The exploratory investigation demonstrates that the tree-based gathering model is effective and productive for malware classification. Additionally, we investigate the adequacy of various ensemble learning procedures to aid the performance and accuracy of the malware recognition. Ensemble model performs in a way that is better than single classifier model in terms of improving the detection accuracy and terms of the apparent multitude of seven evaluative measures. Highlight feature strategy (chi-square) has a lot of effect on the consequences of base learners and that of ensembles This unmistakably distinguishes machine learning algorithms are valuable for classification and clustering of malware samples for given datasets.. The handling time for classification is additionally diminished as a result of eliminating unessential features in feature selection process for classification. In future, we will execute proposed approach on a huge of datasets and will carry out profound investigation in the deep analysis for the classification of packed executables malicious variant, which are viewed as exceptionally undermining in current research.

Table 7. Seven evaluative measures of all the ensembles with their feature selection Methods

Ensembles	Feature Selection Methods	Recall	Precision	F1_Score	False positive rate	ROC	Cohen Kappa	AUC-
Bagging	Chi-square	0.9888	0.9668	0.9776	0.0486	0.9887	0.8623	0.9516
	Without F.S	0.9304	0.9052	0.9176	0.1376	0.8392	0.7302	0.8308
Ada Boosting	Chi-square	0.9850	0.9635	0.9741	0.0534	0.9769	0.8401	0.9250
	Without F.S	0.9187	0.8947	0.9065	0.1531	0.8272	0.7242	0.8180
Gradient Boosting	Chi-square	0.9849	0.9635	0.9741	0.0534	0.9769	0.8401	0.9250
	Without F.S	0.9207	0.8973	0.9088	0.1495	0.8311	0.7107	0.8242
Extreme Gradient Boosting	Chi-square	0.9890	0.9675	0.9781	0.0476	0.9879	0.8565	0.9498
	Without F.S	0.9424	0.9186	0.9303	0.1185	0.8381	0.7281	0.8302
Light Gradient Boosting	Chi-square	0.9888	0.9670	0.9778	0.0483	0.9869	0.8553	0.9471
	Without F.S	0.9338	0.9076	0.9205	0.1339	0.8321	0.7267	0.8272
Majority Voting	Chi-square	0.9259	0.9027	0.9141	0.1417	0.9735	0.8324	0.9129
	Without F.S	0.9062	0.8613	0.8832	0.1951	0.8198	0.7089	0.8091
Random Forest	Chi-square	0.9888	0.9672	0.9779	0.0479	0.9897	0.8628	0.9547
	Without F.S	0.9370	0.9113	0.9240	0.1287	0.8418	0.7312	0.8314
Extra Tree	Chi-square	0.9883	0.9658	0.9769	0.0499	0.9878	0.8568	0.9480
	Without F.S	0.9302	0.9048	0.9173	0.1382	0.8371	0.7298	0.8293

REFERENCES

1. FireEye. (2018 March, 12). The need for speed. [Online]. Available:https://www2.fireeye.com/ismg-incident-response-survey.html.
2. D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford and N. Weaver, "Inside the slammer worm". *IEEE Security & Privacy,* 2003, Vol. l(4), pp. 33–39.
3. G. Chandrashekar, F. Sahin, "A survey on feature selection methods", *Computers & Electrical Engineering*, 2014, Vol.40(1), pp. 16-28.
4. A. Walenstein, M. Venable, M. Hayes, C. Thompson and Lakhotia, "A Exploiting similarity between variants to defeat malware: vilo method for comparing and searching binary programs". In: Proceedings of BlackHat DC, 2007.
5. M. Alazab, "Automated Malware Detection in Mobile App Stores Based on Robust Feature Generation", *Electronics,* 2020, Vol.9, pp. 435-442.
6. E. Seoungyul, L, Hyunjong, K. Donghoon, H. Doosung, "Comparative Analysis of Low-Dimensional Features and Tree-Based Ensembles for Malware Detection Systems", IEEE 2010.
7. A. Eslam and Z. Ivan, "An Ensemble-based Malware-based Malware Detection Model Using Minimum Feature Set", *Mendel,* 2019, Vol. 25(2) pp. 1-10.
8. N. Ochieng, M. Waweru , A. Ismail, " Optimizing Computer Worm Detection Using Ensemble", Hindawi Security and Communication Networks Volume 2019, Article ID 4656480,
9. P. HarshaLatha, R. Mohanasundaram, "A New Hybrid Strategy for Malware Detection Classification with Multiple Feature Selection Methods and Ensemble Learning Methods", *International Journal of Engineering and Advanced Technology (IJEAT*) ISSN, Vol.9(2), pp. 2249 –8958.
10. Ninite. (2019, Nov. 02). Benign data. Available: www.ninite.com.
11. Download. (2019, Nov. 02). Benign data. Available: www.downloads.com.
12. Softpedia. (2019, Nov. 02). Benign data. Available: www.softpedia.com.
13. Totalvirus. (2019, Nov. 02). Online file checker. Available: www.totalvirus.com.
14. Virushare. (2019, Nov. 02). Malware data. Available: www.virushare.com.
15. Virussign. (2019, Nov. 02). Malware data. Available: www.virussign.com.
16. A. Singh and A. Lakhotia, "Game-theoretic design of an information exchange model for detecting packed malware, in Malicious and Unwanted Software (MALWARE)", 2011 6th International Conference on, 2011, pp.1–7.
17. J. Li, K, Cheng, S.Wang, F. Morstatter, R.P. Trevino, J.Tang, H. Liu, "Feature selection: A data perspective", *CM Computing Surveys* (CSUR), 2018, Vol, 50(6) pp. 94-105
18. L. Huiqing, L. Jinyan, and W. Limsoon, "A comparative study on feature selection and classification methods using gene expression profiles and proteomic patterns", *Genome Informatics*, 2002, Vol. 13, pp. 51-60.
19. M.G. Schultz, E. Eskin, F. Zadok, and S.J. Stolfo, "Data mining methods for detection of new malicious executables, in Proc. IEEE Symp. Secur. Privacy, 2001.
20. E. Konstantinou, S. Wolthusen "Metamorphic Virus: Analysis and Detection, Technical Report RHUL-MA-2008-02, Royal Holloway University of London.
21. A.A. Azmee, P.C. Pranto, A.A. Md, D. Orko, I.H. Muhammad, "Performance Analysis of Machine Learning Classifiers for Detecting PE Malware" *International Journal of Advanced Computer Science and Applications*, 2020, Vol. 11(1), pp. 510-517.
22. S. Dreiseitl and L.Ohno-Machado, "Logistic regression and artificial neural network classification models: a methodology review, *Journal of Biomedical Informatics*, 2002, 35(6), pp. 352-359.
23. J.R. Quinlan, "C4.5: programs for machine learning", Morgan Kaufmann Publishers, Inc. 1993.
24. H. Jiawei, K. Micheline, and P.Jian. Data Mining Concepts and Techniques, Elsevier Inc. 2012.
25. H. Sayadi, P.D.Sai, H. Amir, R. Setareh, H.Houman,2018 "Comprehensive Assessment of Run-Time Hardware-Supported Malware Detection Using General and Ensemble Learning. In CF'18, Ischia, Italy.
26. R. Saifur, I. Muhammad, R. Mohsin, M.G.Khawaja, Y. Shumayla, A. Muhammad, "Performance Analysis of Boosting Classifiers in Recognizing Activities of Daily Living Int. J. Environ. Res. Public Health, 2020, Vol.17, 1082-1094.
27. T. Chen and C. Guestrin C, "XGBoost: A Scalable Tree Boosting System. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 785-794.

28. L. Breiman, “Bagging predictors”, *Machine Learning*, 1996, Vol. 24, pp.123-140.
29. J.H. Friedman, “Stochastic gradient boosting”. *Comput. Stat. Data Anal*, 2002, Vol.8 pp. 367–378.
30. G. Ke, Q. Meng, T. Finley, T.Wang, W.Chen , W. Ma , Q. Ye, T. Y.Liu, “LightGBM: A Highly Efficient Gradient Boosting Decision Tree, In Proceedings of the 31st Conference on Neural Information Processing Systems, Long Beach, CA, 2017, USA.
31. Ramnathv and Gdequeiroz. (2017, Sept. 27). Gradient Boosting Machines. Available: https://github.com/ledell/useR-machinelearning-tutorial/blob/master/gradient-boosting-machines
32. B. Jinrong, W. Junfeng, “Improving malware detection using multi-view ensemble learning, *Security Comm. Networks*, 2016, Vol. 9, pp. 4227-4241.
33. B. Peter, “Bagging, Boosting and Ensemble Methods. ETH Zurich, Seminar for Statistik, HG G17, CH-8092, Zurich, Switzerland. 2012.
34. Introduction to Bagging and Ensemble methods.(2010, June). Available: https://blog.paperspace.com/bagging-ensemble-methods.
35. C. Tianqi, G. Carlos, “Xgboost: A scalable tree boosting system”, In Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.
36. K. Sethi, R.Kumar, L. Sethi, P. Bera, P.K.Patra, “A Novel Machine Learning Based Malware Detection and Classification Framework. In 2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security).
37. B.M. Hammas, A. Monemi, J.S. Bassi, I. Ismail, S.M. Nor and M.N. Marsono, “Feature selection and machine learning classification for malware detection, *Jurnal Teknologi*, 2015, Vol.77(1).pp. 234-241.

AUTHORS PROFILE

Fadare Oluwaseun Gbenga, has a Bachelor of Science and Master of Science in Computer Science. He is currently pursuing PhD at the Ekiti State University, Nigeria. He is dynamically engaged in research in the area of cyber-security.

Adetunmbi Adebayo is a Professor in the Department of Computer Science, Federal University of Technology Akure where he obtained his PhD in Computer Science in 2008. He was a recipient of CAS-TWAS postgraduate fellowship at the Institute of Computing Technology, Beijing in 2006 and a Visiting Scholar to Massachusetts Institute of Technology in 2012 under MIT International Science and Technology Initiatives Empowering the Teachers Program. He has several publication in reputable peer-reviewed journals and has also served as reviewers to several peer reviewed journals. His research interests are machine learning, information security and computational linguistics. He is a member of IEEE computer society and Computer professional Council of Nigeria.

Oyinloye Oghenerukevwe Elohor, has a B.Sc., M.Tech., Ph.D in Computer science, her area of research is information security

Mogaji Stephen Alaba, received HND in Physics/Electronics from Federal Polytechnic Ado Ekiti, Nigeria in 1995. He received B.Sc. degree in Computer Science from Joseph Ayo Babalola University Ikeji Arakeji. He also received M.Tech and Ph.D degrees in Computer Science from Federal university of Technology Akure Nigeria in 2012 and 2019 respectively. He is a registered member of Computer Professional of Nigeria (CPN).