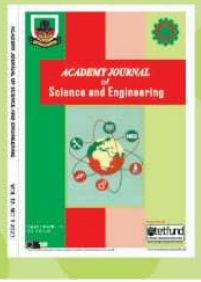

ACADEMY JOURNAL OF SCIENCE AND ENGINEERING
Available online @ www.academyjsekad.edu.ng
AJSE 18 (1) 2024
Original Research

DETECTION OF MALICIOUS WEBSITES USING A THREE MODEL ENSEMBLE CLASSIFIERS

Akolgo, E.A[1], Adekoya, A.F.[2], Korda, D.R.[3], Dapaah, E.O.[4]

[1] Department Computer Science, Regentropfen College of Applied Science, Ghana
[2] Department Computer Science and Informatics, University of Energy and Natural Resources, Ghana
[3] Department of Computing & Information Technology, Bolgatanga Technical University, Ghana
[4] Department of Information and Communication Technology, E.P College of Education, Ghana

Abstract

Malicious attacks are escalating along with the growth of internet users. As a result of that, it is making malware detection inefficient in the cybersecurity field. There are several Machine Learning Classifiers for the detection of malicious websites. Among them include Support Vector Machine (SVM), Random Forest (RF), and Naïve Bayes (NB) which are popularly used techniques. However, when these classifiers are used as stand-alone classifiers, they still suffer from an accuracy sufficiency issue. As a result of that, a three-ensemble classification model to identify a malicious website attack is proposed in this paper to ensure efficient robust malicious detection. Through this paper, it is feasible to reevaluate the malicious attacks and limit the harm that they can cause in the future. In this paper, Support Vector Machine, Random Forest, and Naïve Bayes were combined to develop an ensemble model for malicious website detection. The performance of the proposed ensemble model was evaluated against the three (3) machine learning classifiers using the same malicious and benign websites dataset. Random Forest, Support Vector Machine, Naïve Bayes, and Ensemble models achieved accuracy results of 95.52%, 93.56%, 94.68% and 96.92% respectfully. The results showed that the proposed three ensemble model is a promising solution for malicious website detection.

Keywords: Ensemble; Machine Learning; Support Vector Machine; Random Forest; Naïve Bayes; Malicious Website.

Corresponding author's e-mail: mawugbe4@gmail.com *website: www.academyjsekad.edu.ng*

1.0 INTRODUCTION

Since smartphones are becoming more widely used and there are more Internet-based services available, huge populations are becoming continually more reliant on the web activities like online shopping, banking, paying bills, and engaging in-game activity with pals and random people. These actions have had an impact on the global economy as well as high reliance on financial services offered online (Wang and Huang, 2023) and have raised security concerns for both customers and financial institutions (Sheehan et al., 2019).

Online crime also happens, including spam, fraud, phishing, drive-by exploits and many more, which are illegal centred relating to identity theft. Phishing websites are maliciously designed to look like legitimate websites. Users of the web frequently manage and keep sensitive information that criminals who engage in Internet and Web abuse take advantage of the vulnerability to unjustified gains. Similarly, since many websites are also hosted on cloud servers rather than traditional physical servers, it's imperative to protect websites and user data in the cloud environment. (Hu et al., 2020), (Korda et al., 2021), (Korda et al., 2023)

There is fast expansion on how many websites there are on the internet. The number of webpages available online in 2018 was above 1.6 billion (Internet Live Stats, 2020, Total number of Websites) and it keeps getting bigger over time.

Unfortunately, advancements in technology have led to more sophisticated methods of scamming and harming online consumers. These assaults include phishing websites that promote fake goods, financial scams that trick users into disclosing personal information that can be used to steal their identities or money, and the installation of malware on the user's computer. The top 10 kinds of websites with dangerous content that may harm users are depicted in Figure1. (Softpedia, 2016).

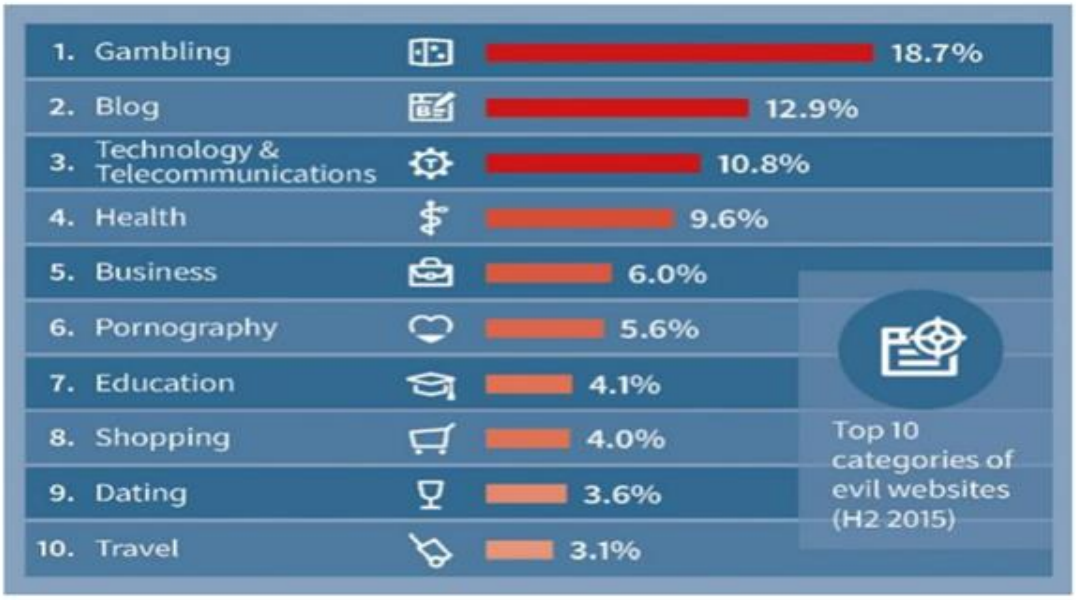

Figure 1: Leading ten categories of malicious websites (Softpedia, 2016).

Web security has become more and more important in recent years as internet connectivity has expanded across the globe. Although growing penetration is great for worldwide communication, it also increases the number of people who have access to websites where they may be exposed to malware, viruses, and other dangerous agents. Therefore, it's more important than ever to identify such websites and prevent ordinary visitors from accessing them (Jang-Jaccard and Nepal, 2014).

Different machine learning algorithms exist. Algorithms for classification are required for harmful website detection. The Random Forest, Support Vector Machine, and Naive Bayes classification methods were used in this study. Additionally, the ensemble approach is employed to increase the algorithms' accuracy. The approach known as ensemble modelling employs a number of algorithms to forecast a result (Karalar et al. 2021). The classification of new test samples is assisted by a mix of several classifiers (Tanha et al., 2020).

One of the more in-depth areas of study for machine learning researchers is ensemble classification (Verma & Mehta, 2017). The ensemble models' majority voting procedure is used. In order to determine which algorithm provides superior classification, Random Forest, Support Vector Machine, Naïve Bayes & Ensemble algorithms were thoroughly tested on fresh data. Using ensemble modelling is encouraged since it lowers generalization errors. Although it is made up of various algorithms, it functions as a single model.

The purpose of this work is to develop an intelligent malicious detection model that evaluates whether malicious activity is taking place on a website using a combination of data mining techniques. The implementation that results from this analysis must be efficient and workable, capable of accurate identification (for example, avoiding false positives and false negatives), and must be capable of accurately alerting the user of the website's malware risk rating.

2.0 LITERATURE REVIEW

A review of the literature on several pertinent works is included in this part. The following categories can be used to group the most popular existing techniques for detecting fraudulent websites:

2.1 Manual Detection of Malicious Websites

One of the popular techniques is to manually search for malicious web pages. The user must be aware of the numerous phishing assaults, and the ability to recognize these websites instantly depends on past knowledge (Mohammad et al., 2015). A simulation of building an examining ACT-R behavioral cognition was developed by (Williams and Li, 2017). In order to develop this ACT-R behavioral cognition, websites were authenticated using the HTTP security signal of a padlock. In Afronz and Greenstadt (2011) a method was developed named "PhishZoo" that employs both site detection procedures using profiling and profile matching. This method compiles an index of all delicate websites, which was compared towards the loaded webpage. The fundamental idea behind this strategy is to compare the content of legitimate and unsafe websites.

Most internet users lack the information necessary to instantly recognize a malicious webpage, which is a disadvantage of the manual detection method. Because people frequently neglect to confirm the credibility of the website while they are preoccupied with their work, even trained individuals become victims of the attack.

2.2 Detection Using URL and Content Features

The many attributes of the website Uniform Resource Locator are used by detection techniques based on URLs to filter malicious websites. Online learning was implemented by (Ma et al., 2009) together with techniques to recognize host-based and lexical characteristics of website URLs. The user-viewed website's content is compared to the original one in content-based detection. A program that recognizes malware by comparing resemblances in website parts was proposed by (Mao et al., 2017) to increase the precision of the prediction of fraudulent websites, this technique employed URL tokens.

The Graph Mining method was used by (hod et al., 2016) to identify malicious sites on the internet. This methodology was able to identify benign websites that are not detectable by the URL analysis method. Moreover, it considers the regular exchanges of information between users and the website. As a result, it develops by analyzing the data as repeated contact between the user and the website, utilizing the AD-URL graph to identify harmful websites.

A reinforcement learning-based method for phishing URL identification was presented by Chatterjee & Namin (2019). They used the IP address, additional URL access requests, and URL metadata to classify URLs as dangerous or benign. Their unique model may compensate based on the actions and state of the learning agents. Their unique model may compensate based on the behavior and condition of the learning agents. They categorized URLs as dangerous or benign based on URL metadata, the IP address of the URL, and extra URL access requests.

Though URL and Content-based capabilities are employed by malicious website recognition, they struggle to identify new website URLs. These techniques lack accuracy and have a moderate false-negative rate (Alsaedi et al., 2022).

Lastly, compared to other feature kinds, such as HTML, JavaScript, visual graphics, and Active X, content-based features have more information that can be collected by crawling the full homepage (Ozker & Sahingoz, 2020).

2.3 Server – Side Detection

For the purpose of identifying phishing websites, Hu, et al., (2016) suggested a method that examines server record data. For resources, the browser makes contact with a trustworthy website. when a user accesses an unauthorized webpage. The genuine website server records this request in the log, which is later used to distinguish between valid and illegitimate ones. A method developed by (Wu, Kuo, & Yang, 2019) uses fuzzy logic in conjunction with machine learning to replace the system's reliance on Boolean algorithms. In the authentication procedure, they employ the site name, the name of a subdomain, with the lifespan and that of the website. Server responses to users' inquiries concerning the legitimacy of the website will be delayed, which is a limitation of these methods. In slow internet connections, they perform inadequately

2.4 Client–Side Detection

The computer code in anti-malicious software may recognize malicious websites and other methods of data access. These frequently notify the user before blocking the content. This category includes applications like anti-virus and anti-malware. In (Armano, et al., 2016) suggested creating a browser add-on or extension to detect malicious websites in real-time. A warning notice appears on the screen if the website is fraudulent after information about the user's online visits is extracted to identify malicious URLs. For the Firefox browser, Marchal, et al., (2017) suggested a real-time browser extension called Off-the-Hook. Off-the-Hook is implemented as a fully-

client-side browser add-on, which preserves user privacy.

2.5 Detection of Malicious Websites through Machine Learning

In order to gather learning data for a machine learning-based method to malicious website detection, feature extraction is necessary. Lexical, host, and content-based aspects are among them; depending on their characteristics, they can be set up in various ways. The information derived from the URL name itself is referred to as lexical-based features. It can be extracted from features like URL string, URL length, length of elements (hostname and top-level domain, for example), and the quantity of special characters and symbols, or it can be derived from the extraction of IP addresses, keywords, and tokens (Aljabri et al., 2022). The URL hostname features can be used to retrieve host-based features, which include details on malicious hosts, their location, how they are identified, and their management style (Wang et al., 2019).

Researchers have developed various methods to classify a particular URL as malicious or benign using feature extraction attributes as a dataset for machine learning. Machine learning algorithm-based classifiers include k-nearest neighbors (KNN), SVM, random forest, naive Bayes, and artificial neural networks. Zhuang et al. (2012) presented an ensemble classifier-based technique for detecting phishing websites. To aggregate the anticipated outcomes from various phishing detection classifiers, they developed an ensemble classification method based on tag information from the HTML attributes. They also used an automatic phishing categorization system based on hierarchical clustering. Vara et al. (2022) suggested a model for detecting fraudulent websites using an SVM classifier. The IP address, the "@" symbol, the "." (dot) symbol, the domain separation using the "–" (underscore or hyphen) symbol, URL redirection, HTTPS token, email subject line, short URL service, hostname length, sensitive words, the number of slashes, Unicode, SSL certificate validity, anchor, iframe, and website ranking are some of the features they took into consideration. They underlined that the secret to raising machine learning models' efficiency is choosing an efficient feature.

Researchers in machine learning concur that how features are chosen and pulled from the web affects how well a model or algorithm performs. Based on the machine learning model's performance, a web can be classified as malicious or benign. Studying the characteristics and machine learning models that are best at correctly identifying fraudulent URLs is therefore essential.

3.0 METHODOLOGY

This paper looks at various machine learning methods with the aim of examining how, a three-model ensemble classification can be used to detect malicious activity on a website. In order to predict harmful websites, a sophisticated three – step ensemble learning model is created. The proposed malicious prediction methodology's general framework is shown in figure 2.

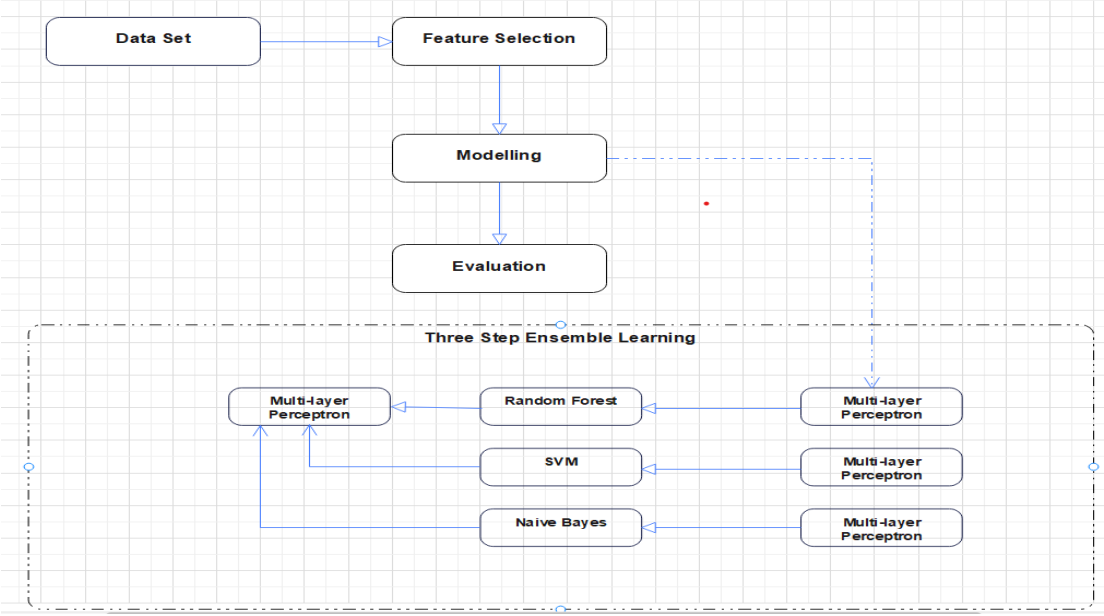

Figure 2: Methodology

3.1 Feature Selection

We picked 20 features and a type label feature from the massive amount of data collected and summarized them in order to create an ensemble classifier. To select qualities that can differentiate between hazardous and benign websites, many websites must be evaluated. To distinguish between reliable and hazardous websites, this step aims to select the most crucial criteria, including URL, Charset, Server, and others.

3.2 Modelling

To handle the most important features that were chosen from the first phase, we created a three-step ensemble framework. Such a framework will be developed using Python, which is utilized for machine learning methods for data mining.

The first step seeks to simultaneously combine three separate multi-layer perceptron neural networks. In the second stage, the Random Forest (RF), Support Vector Machine (SVM), and Naive Bayes (NB) algorithms will be used to process the data from the outcomes of the first stage. A single multi-layer Perceptron neural network will next be used to process the data from the outcomes of the second step.

3.2.1 Random Forest classifier model

Prediction trees make up the random forest classifier. Each tree in the random forest is reliant on random vectors that were randomly sampled and had a comparable distribution to other trees (Devi & Batra, 2023). A random forest is made up of n decision trees, each of which uses the same input but produces a different output. In this instance, the output of the model is taken to be the majority of the outcomes from n decision trees (Hickey, 2007).

The dataset used was sourced from Mendeley Data repository. This data was collected from the Internet by scraping websites using MalCrawle. The dataset with 1,781 rows and 21 columns was used to train this model. In order to create a more powerful model, we employed the SelectKBest algorithm to determine the model's optimal parameters. In order to train the model, K-fold validation was performed. Random forest classifiers are

usually employed for their error generalization technique, when more trees are added to the forest, the random forest's accuracy likewise rises (Jain, 2016). After randomly choosing the attributes for error rate, the accuracy completely depends on the correlation between the trees (Muller & Guido, 2016). Table 1 presents the parameters utilized in the random forests' classification algorithm:

Table 1: Random Forest Experiment Parameters

Parameter	Value
n_estimators	100
max_depth	None
min_samples_split	2
min_samples_leaf	1
max_features	auto
max_leaf_nodes	None
Random state	32
Test size	0.20

3.2.2 Support Vector Machine classifier model

A supervised machine learning technique for creating classifiers is SVM (Hsu et al., 2003). SVM aims to enable label predictions by generating a decision boundary from at least one label, a hyperplane between the two chosen classes, for instance (Balinsky & Blazewicz, 2019).

The data points and the support vectors are handled by the hyperplane (Muller & Guido, 2016). By taking advantage of the space between the data points, it allows for the independent classification of each class (Muller & Guido, 2016). The data points and the support vectors are handled by the hyperplane (Muller & Guido, 2016). By taking advantage of the space between the data points, it allows for the independent classification of each class (Muller & Guido, 2016).

SelectKBest algorithm is used to identify parameters for this model (Desyani et al., 2020).

A set of training examples forms the basis of the SVM training algorithm, or support vectors, each of which is identified as belonging to one of two categories and is treated by the linear classifier as a non-probabilistic binary. The SVM can then be seen as a representation of the models as focuses on space, isolating models in classes divided by a logical gap. More specifically, SVM looks for a hyperplane or set of hyperplanes in a high-dimensional or large-dimensional space. For instance, a great partition is achieved by the hyperplane that is farthest from the closest preparing information purpose of any classification (practical edge), thereby reducing the classifier's speculation error. At that point, unclassified events are mapped onto a comparable space for characterization (Awad et al., 2015). Table 2 describes the experiment parameters of the SVM classifier.

Table 2 SVM experiment parameters.

Parameter	Value
The complexity factor	1.0
Use phrases with lower order	0.001
The number of folds for the internal cross-validation	-1
Number of cross-validation folds	10-fold
Gamma	auto
The random number seed	1
Random state	32
Test size	0.20

3.2.3 Naïve Bayes

Naive Bayes classifiers are a subset of Bayes' Theorem-based classification techniques (Raschka & Mirjalili, 2015). In Naive Bayes classification models, problem instances are represented as vectors of feature values, and the class labels are chosen from a small pool (Microsoft, 2020). In this study, the Gaussian Naïve Bayes Classifier was used because the dataset had a normal distribution. Researchers chose this strategy because naive Bayes guarantees the highest degree of accuracy and makes it simple to analyze the independence between different features.

A set of probabilistic algorithms, including Naive Bayes, uses Bayes' Theorem and current probability theory to predict a text's tag (since often attackers implant malicious tags in the anchor tag of sites). The tag with the highest likelihood is output because they are probabilistic in nature (Sheth et al., 2022). This method is used to determine the likelihood of each tag for text contained within a tag. In order to gather these probabilities, the naive Bayes technique uses the Bayes Principle, which quantifies the likelihood of a feature based on some prior knowledge about conditions that may be relevant to that feature under consideration. Table 3 below presents the experimental parameters of the naïve bayes classifier.

Table 3 Naive Bayes experiment parameters

Parameter	Value
Priors	None
Var_smoothing	1e-09
Random state	32
Test size	0.20

3.3 Evaluation

The evaluation measures the overall efficacy of the suggested classification framework, which will make use of the evaluation metrics for malicious detection issues that are most frequently used, such as classification accuracy, F1 Score, Precision, Recall, ROC, and Area Under Curve (AUC) are used for this paper.

Recall: This indicates how many of the dataset's real positive observations can make accurate predictions. This is often referred to as a model's sensitivity (Naidu et al., 2023).

$$\text{Recall} = \frac{TruePositive}{TruePositive+FalseNegative} \quad \textbf{Equation 1}$$

Precision: This indicates the proportion of expected positive observations that are actually positive (Naidu et al., 2023).

$$\text{Precision} = \frac{TruePositive}{TruePositive+FalsePositive} \quad \textbf{Equation 2}$$

Accuracy: This measures the proportion of accurately anticipated observations, whether they are positive or negative (Naidu et al., 2023).

$$\text{Accuracy} = \frac{TruePositive+TrueNegative}{TruePositive+TrueNegative+FalsePositive+FalseNegative} \quad \textbf{Equation 3}$$

F1 Score: The F1-Score is a metric that concurrently assesses a model's recall and precision. This parameter is essential since it can be difficult to compare models with high recall and low precision to those with low recall and high precision (Naidu et al., 2023). The F1 Score will be used to assess the models trained.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \quad \textbf{Equation 4}$$

3.4 Proposed Model

The suggested model, referred to as ensemble learning, is a machine learning technique that seeks to increase prediction precision by combining the output from multiple models (Chawla, 2017). An ensemble is a collection of hypotheses or learners that are frequently generated from training data using a simple learning technique. There are certain ensemble techniques that use several learning algorithms to create heterogeneous ensembles, but the majority use an individual base learning algorithm, Ensemble approaches create homogeneous ensembles or homogeneous base learners (Mohammed & Kora, 2023).

Using ensemble approaches, many models are created and combined to generate better results. Typically, these methods yield more precise results than a single model would. The many outputs from distinct models are combined to create a single prediction when separate models combine their conclusions. To do this, a vote (or weighted vote) is taken for the classification case, and the average (or weighted average) is calculated for the numerical forecast (Wu & Levinson, 2021).

Support Vector Machine, Random Forest, and Naive Bayes were merged to create an ensemble that will help increase the accuracy of machine learning as shown in figure 3 below.

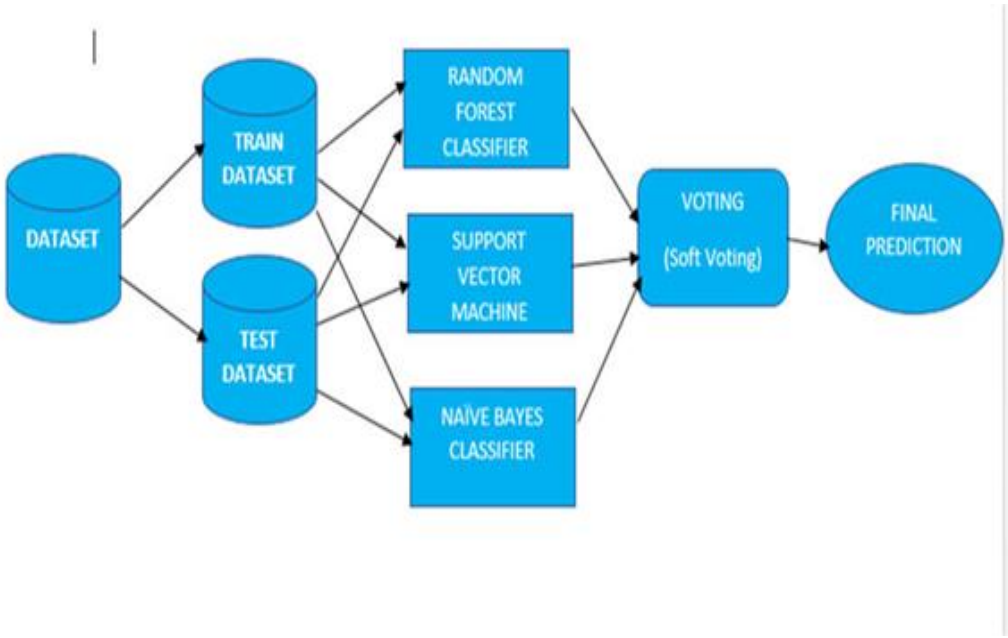

Figure 3: Conceptual framework of the proposed ensemble model

In comparison to using only one model, the ensemble learning method produces greater prediction performance (Chawla, 2017). Combination's fundamental idea is to teach a set of classifiers and give them voting power (Zhou, 2018).

We went over the procedures used to put the Three Classification Ensemble model, which uses the Random Forest Classifier, Support Vector Machine Classifier, and Naive Bayes Classifier, into practice.

The dataset used in this paper was sourced from Mendeley Data repository, which was gathered from Web between November 2019 and March 2020.This data was collected from the Internet by scraping websites using MalCrawle. The dataset consisted of 1,781 rows and 21 columns. There are 4 categorical variables, 15 numerical variables and 2 Date variables. Table 4 gives the list of features or attributes and descriptions of the data used in the malicious and benign dataset.

Table 4 Dataset features and description

Attribute Name	Description
URL	The anonymous identification URL analyzed in the study
URL_LENGTH	The URL length(characters)
NUMBER_SPECIAL_C HARACTERS	The count of the special characters in the URL, for instance (/,%, #, &)
CHARSET	The character encoding standard or character set (categorical variable)
SERVER	The operative system of the server, from the packet response (categorical variable)
CONTENT_LENGTH	The content size of the HTTP header
WHOIS_COUNTRY	The nations we got from the server reaction, explicitly, our content utilized the API of Whois (categorical variable)
WHOIS_STATEPRO	The states we got from the server response, specifically, our script used the API of Whois (categorical variable)
WHOIS_REGDATE	Server registration date. This variable has date values with format (DD/MM/YYY HH:MM)
WHOIS_UPDATED_DA TE	The last update date from the analyzed server
TCP_CONVERSATION_EXCHANGE	The number of TCP packets exchanged between the server and our honeypot client
DIST_REMOTE_TCP_PORT	The number of the ports detected and different to TCP
REMOTE_IPS	Total number of IPs connected to the honeypot
APP_BYTES	Number of transferred bytes
SOURCE_APP_PACKET S	Packets sent from the honeypot to server
REMOTE_APP_PACKE TS	Packets received from server
SOURCE_APP_BYTES	The source of the app bytes
REMOTE_APP_BYTES	The remote app bytes
APP_PACKETS	Complete number of IP created while the correspondence between the honeypot and the server
DNS_QUERY_TIMES	DNS packets generated during the communication between the honeypot and the server
TYPE	Represent the type of web page analyzed (1 is for malicious websites and 0 is for benign websites)

The dataset is typically divided into training and testing portions for the purpose of machine learning models. This is because the model would overfit the data if we used the complete dataset for fitting, which could result in inaccurate predictions (Joseph & Vakayil, 2022). It was split into 70, 30 for training data and testing data respectively. The three (3) models or classifiers each were trained using the dataset to optimize performance. These

three classifiers were combined to form a single classifier using the soft voting technique. By combining these classifiers, a final prediction is then made using the probabilities of each class label.

We performed a series of actions to implement the ensemble model and the steps are detailed below:

Step 1: Pre-processing the data.

The first step taken was cleaning and pre-processing the dataset. This involved removing duplicates, missing values, and outliers. The researcher also standardized the numerical features and converted categorical features to numerical values using one-hot encoding.

Step 2: Feature Selection

In this step, we selected the most relevant features from the pre-processed dataset. This was done using the SelectKBest algorithm, which selects the top K features based on their statistical significance.

Step 3: Model Training

Each (Random Forest, SVM, and Naive Bayes) classifier were trained using the pre-processed data. Hyperparameters were then tuned for each model using cross-validation to optimize performance.

Step 4: Ensemble

Finally, the three classifiers were combined using the soft voting technique. By combining the expected probabilities for each class label, the class label with the highest sum probability is anticipated in the soft voting strategy.

4.0 Results and Discussion

The implementation and assessment results of the proposed algorithm on the dataset are covered in this section, along with an analysis of the outcomes and a comparison of the suggested method with support vector machines (SVM), random forests, and Naive Bayes methods. The paper and its findings are taken into consideration while formulating the paper's question.

We performed an Exploratory Data Analysis (EDA) on the dataset prior to using the various algorithms for machine learning to the data. The main purpose of the EDA was to help have an overview of the data before performing any machine learning modelling with them (Data et al., 2016). It helps identify inconsistencies and errors within the data. Finding outliers or unusual events will help you better grasp the data's patterns and identify outliers. The Multivariate Non-graphic Analysis is the type of EDA used in this paper. Multiple variables combine to form multivariate data (Wegman et al., 2009). In multivariate non-graphical EDA techniques, cross-tabulation or statistics are generally employed to show the relationship between two or more variables of the data (Data et al., 2016).

The first step of the EDA was to import the data. The data used in this experiment was in Comma Separated Value (CSV) format and was imported into the jupyter notebook using the pandas library with Python.

Accuracy

The model's total performance is described by its accuracy. According to the results of the experiment, the ensemble classifier had the highest accuracy, with a score of 96.92%, followed by the Random Forest Classifier with a score of 95.52%, the SVM classifier with a score of 93.56%, the Nave Bayes Classifier with a score of 94.68%

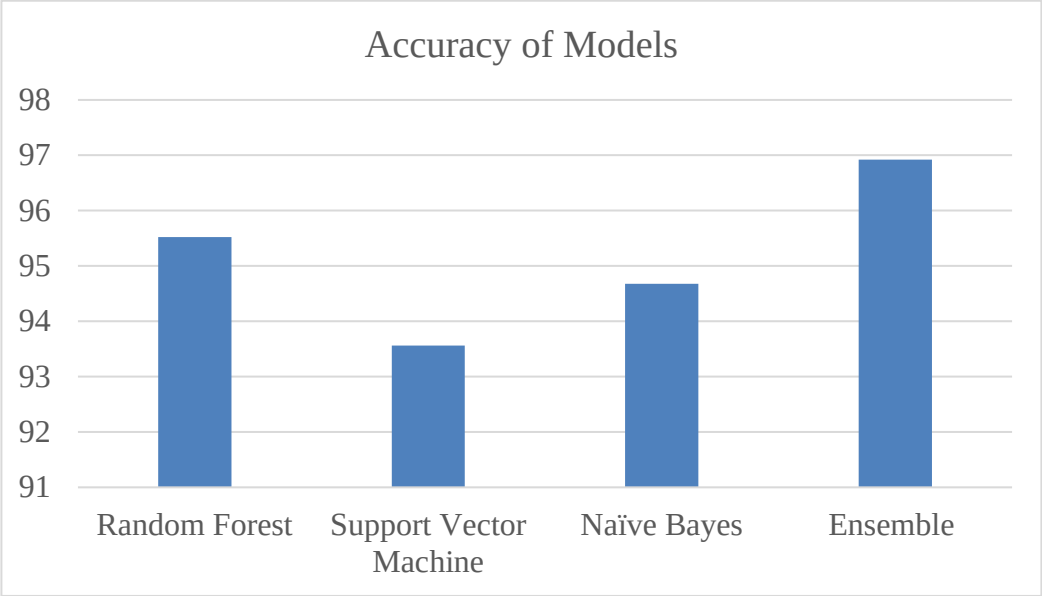

Figure 4: Accuracy Results

Precision

By definition, precision is the proportion of correctly identified positive samples (True Positives) to the total number of correctly or mistakenly classified positive samples (Kuhn & Johnson, 2013). When classifying a machine learning model as positive, precision aids in visualizing its dependability (Kuhn & Johnson, 2013). Precision is mathematically defined as Precision = TP/TP+FP (True Positive/True Positive + False Positive).

In this paper, the Random Forest Classifier, SVM Classifier, and Nave Bayes Classifier all achieved precision values of 0.95, 0.93, and 0.96 respectively. The ensemble classifier achieved a precision of 0.97, which was the highest.

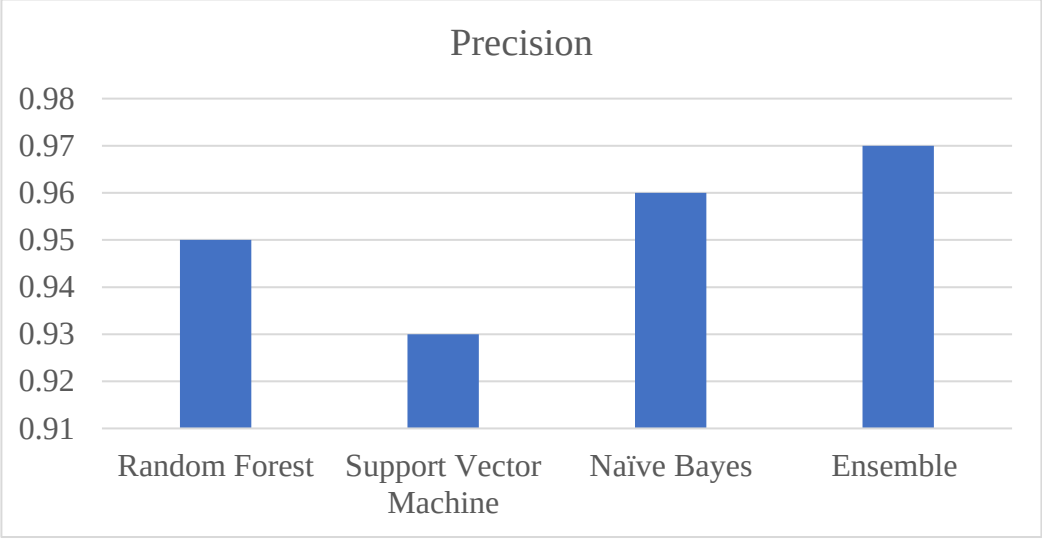

Figure 5: Precision Results

Recall

Recall is the proportion of correctly classified positive samples to the total number of positive samples (Kuhn & Johnson, 2013). A model's recall measures its capacity to identify positive samples (Kuhn & Johnson, 2013). Recall can be mathematically represented as Recall = True Positive / (True Positive + False Negative). A recall of 0.99 was obtained by the Random Forest Classifier, 0.99 was achieved by the SVM Classifier, 0.98 for Naïve Bayes and 0.99 for the Ensemble Classifier.

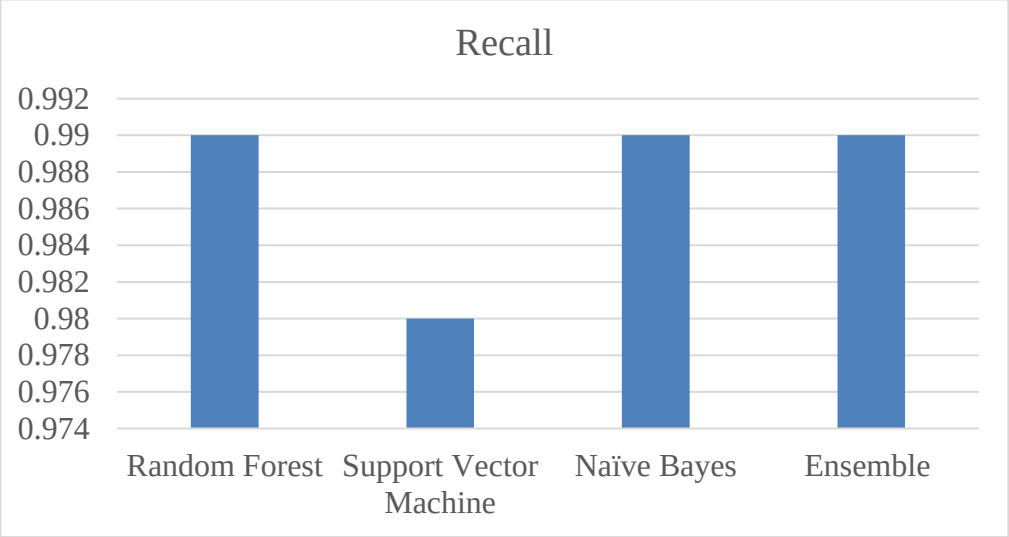

Figure 6: Recall Results

F1 Score

F1 score is an evaluation metric that creates a single metric that combines the precision and recall metrics (Muller & Guido, 2016). The average of precision and recall in the F1 score is computed (Muller & Guido, 2016). Mathematically, the formula for the F1 score is F1 score = 2 x (Precision x Recall) / (Precision + Recall)

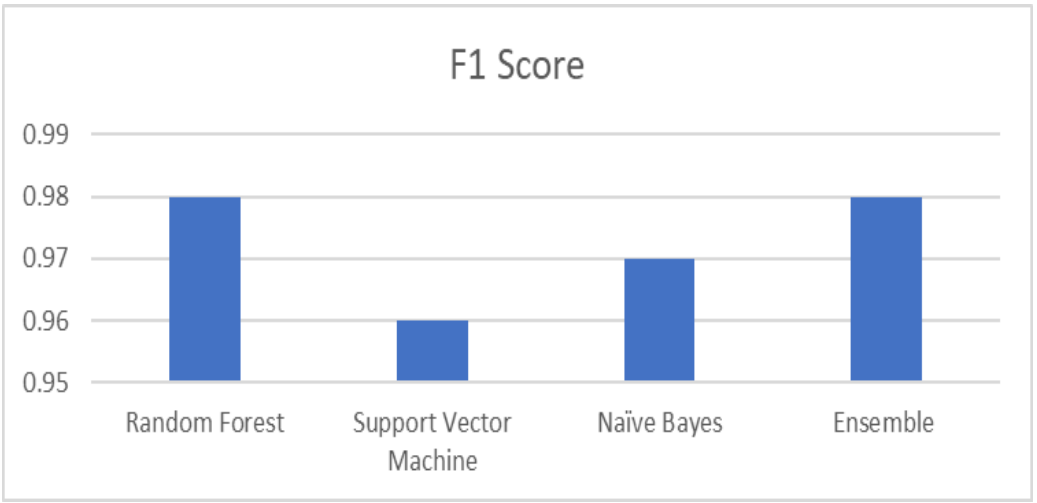

Figure 7: F1 Score Results

Receiver Operating Characteristic (ROC) Curve

Receiver Activity Characteristic plots are used to show how well a binary classifier is doing (Kuhn & Johnson, 2013). It demonstrates how the True Positive Rate (TPR) and the False Positive Rate trade off against one another at different classification levels (FPR) (Kuhn & Johnson, 2013).

The following figures show the ROC of the models:

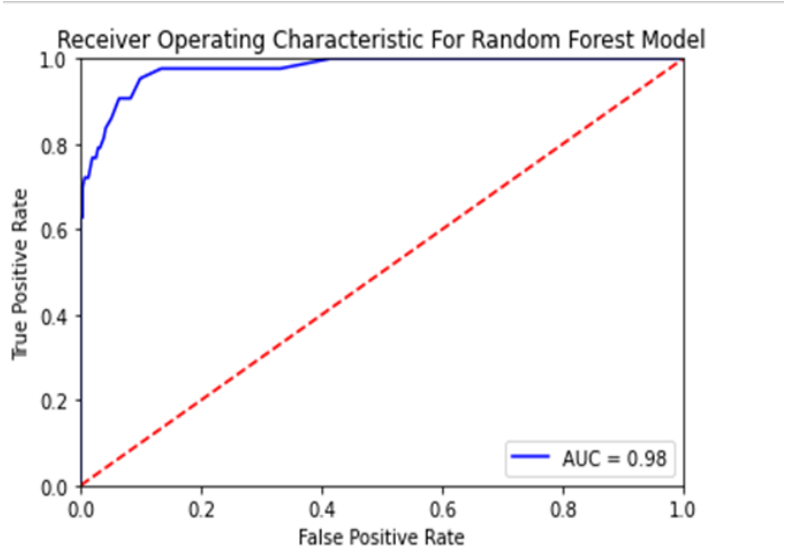

Figure 8: ROC for Random Forest Model

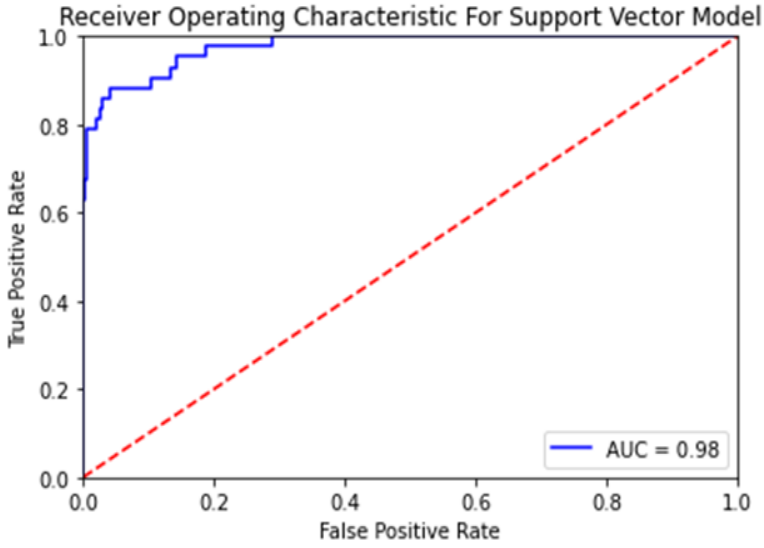

Figure 9: ROC for SVM Model

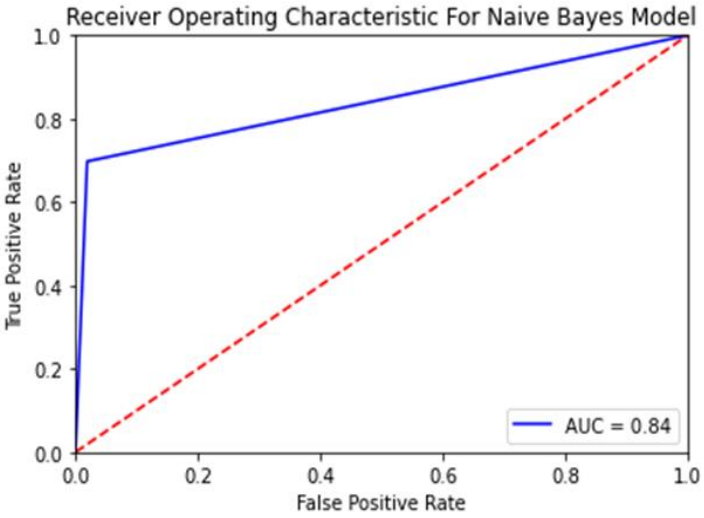

Figure 10: ROC for Naïve Bayes Model

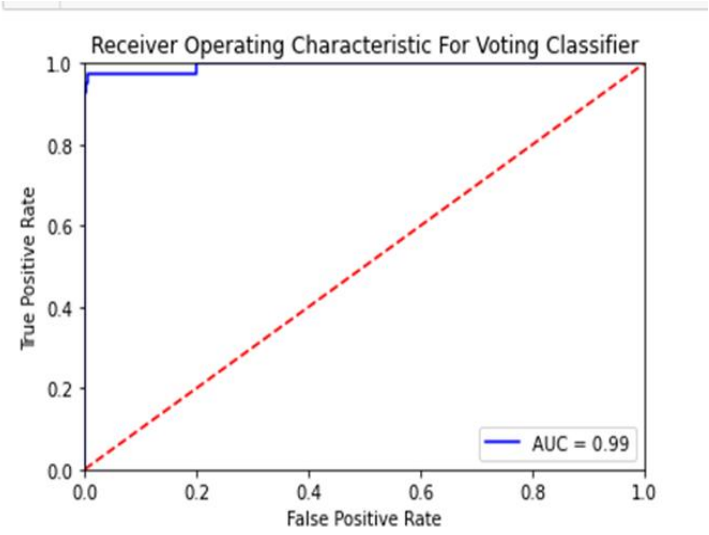

Figure 11: ROC for Ensemble Classifier

Comparative Analysis

This paper demonstrated that the ensemble model outperformed the other models in terms of accuracy, a measure of a model's overall performance.

Accuracy

In this paper, it was demonstrated that the ensemble model performed better than the other models in terms of accuracy, a metric used to assess a model's general performance. The outcomes of this study show a claim that the ensemble model is most accurate at detecting malicious websites, outperforming the Random Forest, Naive Bayes, and SVM classification models. This finding has significant implications for the field of cyber security, as the ability to accurately detect

malicious websites is important for safeguarding individuals and organizations from cyber-attacks such as denial of services attacks or distributed denial of service attacks (Korda & Dapaah, 2023).

Precision

The results of this study demonstrate that an ensemble model is the most precise at detecting malicious websites when compared to the Random Forest, Naive Bayes, and SVM classification models.

All models achieved a high precision score of more than 0.90, indicating a high level of effectiveness in malicious websites. However, the ensemble model's precision score exceeded the other models by a notable margin. One potential explanation for the ensemble model's superior precision is its ability to integrate the predictions of multiple models (Zhou, 2018). By combining the strengths of multiple models, the ensemble model generates a final prediction that is more robust and stable (Zhou, 2018).

Recall

The results of this study show that the ensemble model, Random Forest, and Naive Bayes classification models achieved a recall of 0.99 for detecting malicious websites, while the SVM classification model achieved a recall of 0.98. Recall is a measure of how well-equipped a model is to correctly determine all occurrences of a particular class, in this case, malicious websites.

F1 Score

The findings of this investigation demonstrate that the ensemble model and the Random Forest classification models obtained a 0.98 F1 score for detecting malicious websites, while the Naive SVM classification model received an F1 rating of 0.96 while the Bayes model received an F1 rating of 0.97. F1 rating is a measure of a model's performance that takes both recall and precision into account (Kuhn & Johnson, 2013).

A high F1 score indicates that a model is able to identify a high proportion of real positives (i.e., accurately identify malicious websites) while minimizing how many false positives there are (i.e., benign websites incorrectly identified as malicious).

It is worth noting that the F1 scores of the models were relatively high, but not perfect. This suggests that there is room for improvement in the ability of these models to detect malicious websites, and further research could be conducted to identify ways to increase F1 scores. This could include incorporating additional features into the models or using more advanced deep learning strategies for machine learning models.

ROC Curve

The outcome of this study shows that the ensemble model outperforms the Random Forest, Naive Bayes, and SVM classification models in terms of ROC and AUC for detecting malicious websites. A plot of true positive rate vs. false positive rate is known as the receiver operating characteristic, or ROC (Kuhn & Johnson, 2013). Area Under the Curve (AUC), which measures a model's overall performance; a larger AUC denotes a better model. The effectiveness of a model's capacity to distinguish between several classes, in this example dangerous and benign websites, can be evaluated using these assessment criteria.

It is worth noting that the ROC and AUC scores of all four models were relatively high, with all models achieving an AUC above 0.9. This suggests that all four models are effective at detecting malicious websites, and the choice of model may depend on other factors such as computational complexity and the specific requirements of the application.

5.0 CONCLUSION

Using three integrated detectors—Random Forest, SVM, and Naive Bayes—we examined the issue of harmful website identification in this paper. These three detectors had somewhat different results, but they all had accuracy scores that were lower than the combined ensemble when it came to applying it to the problem of detecting malicious websites. Using a dataset, the proposed model is put into practice and assessed.

The methods employed in this paper involved training an ensemble classification model for detecting malicious activity in a website. By employing this technique in training a model, malicious websites are recognized at approximately a 97% accuracy rate.

In this paper, a built model for detecting malicious has been developed using an ensemble, based on three algorithms. Based on this, it can be concluded that the objective, which was to develop a modified ensemble model for malicious website detection using three model ensemble classification algorithms has been achieved.

With this ensemble model, it has the chance to detect malicious activities on websites for an action to be taken by the user. The researchers, therefore, recommend that this proposed model be adopted by blacklist providers, to help them achieve a better prediction of malicious activities or webpages.

6.0 FUTURE WORK

In future, we are certain that, there are still a lot of techniques that can be implemented to improve the ensemble model. Some of the future works that can be considered are:

- to investigate how well machine learning algorithms can detect rogue websites in the face of more recent hostile activities.
- to create a browser extension that will instantly warn you when you visit suspicious websites.
- The model's precision can be increased to perform better in terms of reliably predicting harmful websites.

REFERENCES

Aljabri, M., Alhaidari, F., Mohammad, R. M. A., Mirza, S., Alhamed, D. H., Altamimi, H. S., & Chrouf, S. M. (2022). An assessment of lexical, network, and content-based features for detecting malicious urls using machine learning and deep learning models. *Computational Intelligence and Neuroscience*, *2022*.

Wang, H. H., Yu, L., Tian, S. W., Peng, Y. F., & Pei, X. J. (2019). Bidirectional LSTM Malicious webpages detection algorithm based on convolutional neural network and independent recurrent neural network. *Applied Intelligence*, *49*, 3016-3026.

Ozker, U., & Sahingoz, O. K. (2020, September). Content based phishing detection with machine learning. In *2020 International Conference on Electrical Engineering (ICEE)* (pp. 1-6). IEEE.

Zhuang, W., Jiang, Q., & Xiong, T. (2012, June). An intelligent anti-phishing strategy model for phishing website detection. In *2012 32nd International Conference on Distributed Computing Systems Workshops* (pp. 51-56). IEEE.

Chatterjee, M., & Namin, A. S. (2019, July). Detecting phishing websites through deep reinforcement learning. In *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)* (Vol. 2, pp. 227-232). IEEE.

Vara, K. D., Dimble, V. S., Yadav, M. M., & Thorat, A. A. (2022). Based on URL Feature Extraction Identify Malicious Website Using Machine Learning Techniques. *International Research Journal of Innovations in Engineering and Technology*, *6*(3), 144.

Afronz, S., & Greenstadt, R. (2011). Dettecting Phishing Attacks by looking at them. *IEEE Fifth International Conference on Semantic Computing.* Palo Alto.

Armano, G., Marchal, S., & Asokan, N. (2016). Real-time Client-side Phishing Prevention add-on. *IEEE 36th Conference on Distributed Computing.*

Futai, Z., Yuxiang, G., Bei, P., Li, P., & Linsen, L. (2016). Web Phishing Detection Based on Graph Mining. *2nd IEEE International Conference on Computer Communications.*

Hodowu, D. K., Korda, D. R., & Ansong, E. (2020). An Enhancement of Data Security in Cloud Computing with an Implementation of a Two-Level Cryptographic Technique, using AES and ECC Algorithm. *International Journal of Engineering Research & Technology*, *09*(09).

Hu, J., Zhang, X., Ji, Y., Yan, H., Ding, L., Li, J., & Meng, H. (2016). Detecting Phishing Websites Based on the Study of the FInancial Industry Webserver logs. *3rd International Conference on Information Science and Control Engineering.*

Wang, Z., & Huang, X. (2023). Understanding the role of digital finance in facilitating consumer online purchases: An empirical investigation. *Finance Research Letters*, 103939.

Sheehan, B., Murphy, F., Mullins, M., & Ryan, C. (2019). Connected and autonomous vehicles: A cyber-risk classification framework. *Transportation research part A: policy and practice*, *124*, 523-536.

Cremer, F., Sheehan, B., Fortmann, M., Kia, A. N., Mullins, M., Murphy, F., & Materne, S. (2022). Cyber risk and cybersecurity: a systematic review of data availability. *The Geneva Papers on risk and insurance-Issues and practice*, *47*(3), 698-736.

Karalar, H., Kapucu, C., & Gürüler, H. (2021). Predicting students at risk of academic failure using ensemble model during pandemic in a distance learning

system. *International Journal of Educational Technology in Higher Education*, *18*(1), 63.

Tanha, J., Abdi, Y., Samadi, N., Razzaghi, N., & Asadpour, M. (2020). Boosting methods for multi-class imbalanced data classification: an experimental review. *Journal of Big Data*, *7*, 1-47.

Mohammad, R. M., Thabtah, F., & McCluskey, L. (2015). Tutorial and critical analysis of phishing websites methods. *Computer Science Review*, *17*, 1-24.

Jang-Jaccard, J., & Nepal, S. (2014). A survey of emerging threats in cybersecurity. *Journal of computer and system sciences*, *80*(5), 973-993.

Mao, J., Tian, W., Li, P., Wei, T., & Liang, Z. (2017). Phishing-alarm: Robust and efficient phishing detection via page component similarity. *IEEE Access*, *5*,17020-17030.

Alsaedi, M., Ghaleb, F. A., Saeed, F., Ahmad, J., & Alasli, M. (2022). Cyber threat intelligence-based malicious url detection model using ensemble learning. *Sensors*, *22*(9), 3373.

Naidu, G., Zuva, T., & Sibanda, E. M. (2023, April). A Review of Evaluation Metrics in Machine Learning Algorithms. In *Computer Science On-line Conference* (pp. 15-25). Cham: Springer International Publishing.

Wu, H., & Levinson, D. (2021). The ensemble approach to forecasting: a review and synthesis. *Transportation Research Part C: Emerging Technologies*, *132*, 103357.

Devi, U., & Batra, N. (2023, February). Comparison of Decision Tree and Random Forest for Default Risk Prediction. In *International Conference On Innovative Computing And Communication* (pp. 147-155). Singapore: Springer Nature Singapore.

Hickey, R. J. (2007). Structure and Majority Classes in Decision Tree Learning. *Journal of Machine Learning Research*, *8*(8).

Desyani, T., Saifudin, A., & Yulianti, Y. (2020, July). Feature selection based on naive bayes for caesarean section prediction. In *IOP Conference Series: Materials Science and Engineering* (Vol. 879, No. 1, p. 012091). IOP Publishing.

Awad, M., Khanna, R., Awad, M., & Khanna, R. (2015). Support vector machines for classification. *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*, 39-66.

Sheth, V., Tripathi, U., & Sharma, A. (2022). A Comparative Analysis of Machine Learning Algorithms for Classification Purpose. *Procedia Computer Science*, *215*, 422-431.

Mohammed, A., & Kora, R. (2023). A comprehensive review on ensemble deep learning: Opportunities and challenges. *Journal of King Saud University-Computer and Information Sciences*.

Joseph, V. R., & Vakayil, A. (2022). SPlit: An optimal method for data splitting. *Technometrics*, *64*(2), 166-176.

Balinsky, A. P., & Blazewicz, J. G. (2019). Support Vector Machines: Algorithm, Insights and Applications. *Springer*.

Chawla, N. (2017). Ensemble Machine Learning: Improving Predictive Performance. *Springer*.

Hodowu, D. K., Korda, D. R., & Ansong, E. (2020). An Enhancement of Data Security in Cloud Computing with an Implementation of a Two-Level Cryptographic Technique, using AES and ECC Algorithm. *International Journal of Engineering Research & Technology*, *09*(09).

Hsu, C. W., Chang, C. C., & Lin, C. J. (2003). A practical guide to support vector classification.

Jain, A. (2016). A complete tutorial on ridge and lasso regression in python. *Analytics Vidhya*, *28*.

Korda, D. R., & Dapaah, O. E. (2023). The Role of Cyberattacks on Modern Warfare: A Review. *International Journal of Research and Innovation in Applied Science, VIII*(VII), 286-292.

Korda, D. R., Akolgo, E. A., Dapaah, E. O., & Hodowu, D. K. (2023). Securing Data in Clouds using the SDC Algorithm: Current Trends and Research Directions. *International Journal of Computer Applications*, 23-28.

Korda, D. R., Ansong, E., & Hodowu, D. K. (2021). Securing Data in the Cloud using the SDC Algorithm. *International Journal of Computer Applications, 183*(25), 24-29.

Kuhn, M., & Johnson, K. (2013). *Applied predictive modeling* (Vol. 26, p. 13). New York: Springer.

Mao, J., Tian, W., Li, P., Wei, T., & Liang, Z. (2017). Phishing-Alarm: Robust and Efficient Phishing Detection via Page Component Similarity. *IEEE Access, 5*, 17020-17030.

Marchal, S., Armano, G., Grondahl, T., Saari, K., Singh, N., & Asokan, N. (2017). Off-the-hook: An Efficient and Usable Client-Side Phishing Prevention Application. *IEEE Trans Computing*, 1717-1733.

Microsoft . (2020). *Machine Learning Algorithms*. (Microsoft Azure) Retrieved from https://azure.microsoft.com/en-gb/overview/machine-learning-algorithms/#techniques

Muller, A., & Guido, S. (2016). *Introduction to Machine Learning with Python: A Guide for Data Scientists.* O'Reilly Media, Inc.

Data, M. C., Komorowski, M., Marshall, D. C., Salciccioli, J. D., & Crutain, Y. (2016). Exploratory data analysis. *Secondary analysis of electronic health records*, 185-203.

Raschka, S., & Mirjalili, V. (2015). *Python Machine Learning.* Packt Publishing Ltd.

Softpedia. (2016). *More than Half of the World's Malicious Websites Are Hosted in the US*. (Softpedia) Retrieved March 4, 2022

Wegman, E. J., Said, Y. H., & Scott, D. W. (2009). Introducing WIREs Computational Statistics. *Wiley Interdisciplinary Reviews: Computational Statistics*, *1*(1), 1-2.

Verma, A., & Mehta, S. (2017, January). A comparative study of ensemble learning methods for classification in bioinformatics. In *2017 7th International Conference on Cloud Computing, Data Science & Engineering-Confluence* (pp. 155-158). IEEE.

Williams, N., & Li, S. (2017). Simulating Human Detection of Phishing Websites: An investigation into the applicability of the ACT-R cognitive behaviour architecture model. *2017 3rd IEEE International Conference on Cybertics.*

Zhou, Z. H. (2018). Ensemble Machine Learning: Methods and Applications. *Springer*.

Wu, C., Kuo, C., & Yang, C. (2019). A Phishing Detection System based on Machine Learning . *International Conference on Intelligent Computing and its Emerging Applications.* Tainan, Taiwan.