Journal of Engineering Science and Technology
Special Issue on ICIT2022 Vol. 19, No. 1 (2024) 95 - 108

DEEP LEARNING AND MACHINE LEARNING APPROACHES FOR E-MAIL PHISHING DETECTION

MANOHARAN THIYAGARAJAN[1],
DURAI MUTHASARAN[1], HARPRITH KAUR RAJINDER SINGH[2]

[1]AMETUniversity, Department of Computer Science and Engineering, Chennai, India
[2]INTI International University, Faculty of Data Science and Information Technology, Nilai, Malaysia
Corresponding author: harprith.randhawa@newinti.edu.my

Abstract

The advancements in Artificial Intelligence (AI) have been phenomenal, leading to more sophisticated software and autonomous robots. Parallel to this, the cyber world has evolved into a battlefield where access, influence, security, and power are fought for. This article will discuss important areas of AI, such as machine learning, to clarify their function in cyber security and the ramifications of these emerging technologies. In this article, we will explore and highlight the many machine-learning technologies that may be used to improve cyber security. The increasing number of people who have access to the internet has made phishing attacks increasingly dangerous. People's everyday lives and the internet ecosystem are increasingly at risk from phishing attacks. Account passwords, credit card numbers, and other personal information may be stolen in these types of attacks since the attacker poses as a legitimate business or government agency. Spoofed websites, also known as phishing websites, are designed to look and sound like legitimate websites to steal users' login credentials. Thus, in this work, we will explore deep learning and machine learning techniques and execute all of them on our dataset, selecting the optimal method with the highest precision and accuracy for fraudulent webpage identification. As a result of the research, we may have better tools to prevent future phishing assaults. In this study, we compare the effectiveness of several feature sets and classification algorithms in detecting new phishing emails and detecting variations in the quantity of authentic data. To evaluate the current methods, a new dataset is constructed. We used the Logistic Regression (LR), K Nearest Neighbour (KNN), Decision Tree (DT), Random Forest (RF), XG Boost, and AdaBoost techniques to produce a feature-extracted comma-separated values (CSV) file and label file. In this experiment, we model the detection of a hacked email account as a classification problem. Comparative analysis and practical application indicate that Logistic Regression (LR), K Nearest Neighbour (KNN), Decision Tree (DT), Random Forest (RF), XG Boost, and AdaBoost algorithms perform better and more accurately in identifying phishing emails.

Keywords: AdaBoost, Artificial intelligence, Decision tree, K nearest neighbour, Logistic regression, Phishing attacks, Process innovation, Random forest, XG Boost.

1. Introduction

Big Data, cloud computing, artificial intelligence, etc., were bandied around in various contexts over and over again, sometimes without a firm grasp on their relevance or how they might be successfully used to solve real-world issues. Artificial intelligence (AI) is the development of smart computers with the ability to acquire information from experience and mimic human performance and behavior. With the use of this technique, computers can be taught to analyse massive datasets in search of patterns. Due to its unique qualities, such as flexibility, scalability, and the capability to swiftly adapt to new and unexpected obstacles, machine learning methods have been employed in many fields of research. The rapid growth of social media networks, cloud and internet applications, online banking, the mobile environment, the smart grid, etc. has made cybersecurity an increasingly important area of study. Machine learning, a subfield of artificial intelligence, is being used to effectively address some of the issues (see Fig. 1). As a potent kind of Artificial Intelligence (AI), Machine Learning is a valuable resource for the cybersecurity industry. Applied AI (AI fuelled by Machine Learning) is becoming an increasingly significant tool for automating the identification and categorization of malware at scale.

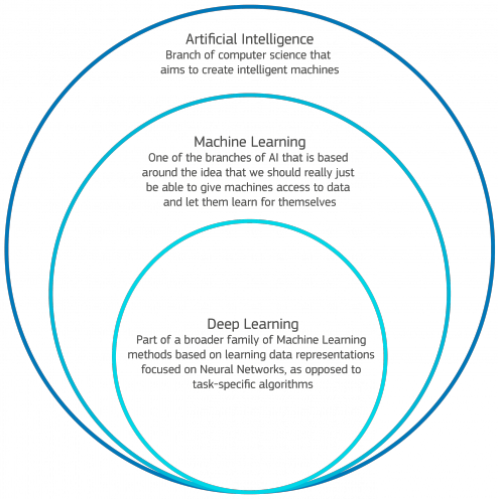

Fig. 1. Artificial intelligence fields.

The rise of phishing attacks is among the most worrying trends in cybercrime [1]. Intentional phishing attempts have increased in frequency and success over the last several years [2], targeting people who would go online to take advantage of the services provided by the internet. Spoof websites are created by attackers to trick users into divulging personal information [3]. This information includes the user's login, password, card number, and other private and sensitive data. In turn, these phishing assaults generate profit for the perpetrators. Online banking, online payment systems, and online shopping carts are common targets [4] for these kinds of assaults. Our polling indicates that blacklist technology was first used to counteract phishing attempts [5]. Despite its success, the blacklist system may be gamed by an attacker who just changes certain elements in the URLs to make it seem like a valid resource and prevent the system from flagging the site as a phishing campaign. As a bonus,

attackers have discovered a way to win over visitors by making them believe they are on a highly secure website by asking them security questions [6]. Users are readily duped by the phishing website's questions and answers.

Consequently, these assaults may be averted by the detection of such phony or fake websites and the cultivation of vigilance among users. Phishing websites may be detected with the use of machine learning algorithms, which is a powerful method [7]. Several models, including one with a deep learning model, are examined, and the one with the highest accuracy is chosen for detecting phishing websites. To combat phishing through email, we use a variety of machine learning and deep learning techniques, including Logistic Regression (LR), K Nearest Neighbour (KNN), Decision Tree (DT), Random Forest (RF), XG Boost, and AdaBoost. The suggested characteristics are simply parsed out of the various components of an email (header, body, text, and connections) using this method. To get them, you need neither an active internet connection nor the help of any outside authorities or other systems. In this project, we use the online CSDMC SPAM database to compile spam and non-spam email datasets. We use email datasets for both training and testing, and we extract features from all of the.eml (email) datasets. We next transform the extracted dataset into a CSV format. After importing the CSV and label file, NLP is used to determine whether or not the spamming or otherwise spam dataset has any errors before applying text processing to the dataset. Following this, we use several different algorithms, including Logistic Regression (LR), K Nearest Neighbour (KNN), Decision Tree (DT), Random Forest (RF), XG Boost, and AdaBoost, to categorize the email phishing assaults. Results are calculated by our model utilizing a variety of ML and DL approaches. But the execution time and system report are superior for the Logistic Regression (LR), K Nearest Neighbour (KNN), Decision Tree (DT), Random Forest (RF), XG Boost, and AdaBoost algorithms. With these findings, we can show the effectiveness of our features against spam email sites and assess them in comparison to current methods for identifying malicious emails.

1.1. Machine learning

Synthesizing and analysing computational entities that behave intelligently is the focus of Artificial Intelligence [Wan 08], an area of study. As a subset of AI, machine learning is becoming more important. There are currently just a few uses for AI, and they all have to do with Machine Learning (ML). As processing power, storage capabilities, and data collecting expand, the dots between Artificial Intelligence and Machine Learning [Mar 18] are being drawn more broadly than ever before across industries and applications.

The term "Machine Learning" refers to the process of instructing a computer to solve a problem or make a choice without human intervention. This is in contrast to the more conventional method of programming, which is providing detailed instructions to a computer so that it may carry out desired tasks such as answering questions. In order to account for every eventuality, it is necessary to pre-program every conceivable scenario. Statistics, mathematical optimizations, and data mining are all examples of methods that might fall under the ML umbrella. Machine learning algorithms attempt to reason about their actions and discover solutions to issues by inferring these properties from models trained on data samples representing realistic situations. Different ML variants have quite distinct purposes. Figure 2 depicts the three basic categories of ML that may be defined by taking a broad view of the field: supervised learning, unsupervised learning, and reinforcement learning.

1.2. Supervised learning

For the purposes of supervised learning, the computer is taught by being exposed to labelled samples of data. An input variable and also an output variable Q is utilized in a supervised learning technique to generate and train a transformation matrix (f) from P to Q. Each fresh input (P) should result in a unique anticipated output (Q), and this is the purpose of supervised learning algorithms. To put it another way, the learning algorithm is given a collection of inputs and their associated outputs, and the algorithm learns by making analogies between its own output and the proper outputs in an effort to identify faults and have the learning model adjusted appropriately. In order to estimate the label values for unlabelled data, supervised learning algorithms look for patterns. Classification, regression, prediction, and similar techniques do this. The goal is to have the machine assess fresh data, and find the solution, based on the training it has received. Clinical diagnostics and voice recognition are only two examples of areas where supervised learning has proven useful.

1.3. Unsupervised learning

Training a system using data that doesn't contain labels is called unsupervised learning. When just a set of input data is supplied, without any corresponding output variables, this is known as unsupervised learning. With unsupervised learning, you're supposed to model the data's architecture to figure out what it is all about. To draw conclusions from data, algorithms must first find patterns, make inferences, and extract meaning. Contrary to supervised algorithms, these ones don't rely on training data to anticipate results. To put it another way, the computer has no idea what ever the on what or how to interpret the results. If the machine is to produce the desired result, it must first determine the structure and patterns of the incoming data on its own. The categorization of film genres by Netflix is an application of unsupervised learning.

1.4. Reinforcement learning

Machines use reinforcement learning to learn how to successfully navigate their surroundings. Like unsupervised learning, the computer is taught on unlabelled data. In contrast, reinforcement learning involves the machine being provided with information on its performance.

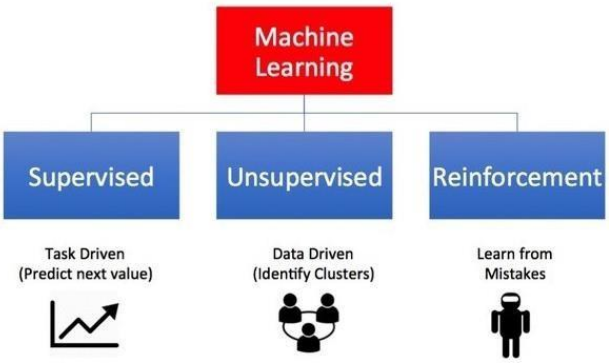

Fig. 2. Three types of machine learning.

1.5. Deep learning

With the help of examples provided by humans, deep learning helps computers to learn. Driverless vehicles rely heavily on deep learning technology, which allows them to do things like identify stop signs and tell people apart from lampposts. It is the driving force behind voice-activated interfaces on Smartphone's, iPads, Televisions, and hands-free audio systems. There is a lot of interest in deep learning right now, as well as for excellent purpose. That which was formerly impossible has now been accomplished. To execute categorization tasks, deep learning trains a computer model using data such as photos, text, or audio. For certain tasks, deep learning models may even outperform humans. To properly train a model, it is common practice to use a vast quantity of data sets and a multi- layered neural network design.

1.6. Cyber security

When it comes to the hardware and software utilized in the manufacturing process, referred to as industrial IT and Operations Technologies (OT), security is quickly becoming one of the most pressing concerns. Information systems, networks, software, and data are all vulnerable to attacks from cybercriminals, terrorist organizations, and hackers who get access to the Internet via a variety of exploits. The term "cyber security" refers to the practice of guarding your digital devices and data stored on the internet and other networks against intrusion. The ever-changing nature of security threats is one of cyber security's biggest obstacles. Viruses, worms, and Trojan horses were the primary cyber dangers to the business network, which was built on mainframes, a client-server approach, and a closed set of computers. Malware designed to cause harm to computer systems (such as viruses, worms, and Trojan horses) was the primary emphasis. Threat actors might sometimes go for random Internet-connected PCs. As a consequence, the security execution can be expanded, and the defence system against the growing number of advanced cyber threats may be improved with the help of AI. Intelligent agents, neural networks, expert systems, data mining, machine learning, and deep learning are just some of the AI methods that may be used to cyber security.

2. Related Works

Several more publications in the field use machine learning methods to the problem of spotting phishing sites. [8] They've detailed a literature review on detecting phishing websites that focuses on security issues. Scientists have used a wide variety of techniques for identifying phishing domains. These methods move away from traditional blacklist/ whitelist approaches and instead rely on machine learning. Notable prototypes of machine learning methods used in detection are shown. The vast majority of earlier systems relied only on supervised algorithms[9]. Online education and deep learning, two promising methods, have not been well studied.

Nowadays, phishing assaults pose a serious risk to people and teams inside businesses. Phishing is a technique used to steal sensitive information from network users by tricking them into visiting a bogus website. Due to its better classification skills for particular datasets and its active learning capabilities, backpropagation (BP) neural architecture is an important heuristic ML approach in phishing site detection systems[10]. Affecting the BP neural connection through into regional minimum and slow learning federation is the incorrect identification of starting boundaries for

example the underpinning weight and edge. This study suggests DF as a solution to these problems. Based on the proposed BP neural connection and a double component assessment framework [11], the GWO-Back propagation neural network (BPNN) is a convincing phishing site reveal model. The accuracy with which phishing websites are identified is enhanced by the two-stage review procedure. The methodology is distinct from two other approaches to revealing phishing sites. The results of our tests have shown the accuracy and robust adaptability of our model.

In Latin America, phishing attempts have increased beyond the capacity of network security professionals to counter. Big data, machine learning, and analysis of information are proposed solutions in the intellectual security problem [12] to speed up the time it takes to identify attacks. In this study, we look analyse how ML methods are utilized to eliminate the problem of aberrant behavior connected to phishing online assaults. Here, machine learning (ML) is used to the problem of identifying phishing attempts by analysing URLs (URLs).

To put it simply, cloud computing is a new kind of information technology (IT) that allows customers to access a shared pool of configurable, virtualized, on-demand computing resources with little upfront investment and ongoing operational expenses. Board associations have compiled and made available these materials online, following established protocols and guidelines. Both new developments and the system's history reveal weaknesses that might be exploited by malicious actors. As previously mentioned, DDoS assaults are among the most effective for causing verified damage and affecting cloud performance [13]. This might waste a lot of time on the server or significantly reduce the efficiency of transferring data associated with cloud structures [14]. For this reason, in this research, we built a DDoS region architecture using the C.4.5 algorithm to mitigate the DDoS risk. In order to verify our system, we picked many different ML algorithms and studied the results. tools like lection and python for ML research and development.

Social engineering attacks like phishing target unsuspecting users in an attempt to get private information [15]. This study makes a novel system proposal, the finest program design for customers to accurately recognize phishing websites.

The URL-based attributes or features are removed from this system in accordance with the principle of extraction construction. The client side executes these cycles with the aid of a tailored process plan [14]. Even while ML frameworks for phishing objection recognition have been the subject of recent study, they are not yet in a usable form for those who lack this specialized information. In this article, we provide a new technique for securing the present user experience while striving to improve security, which we call the 'Embedded Phishing Detection Browser' (EPDB). We has developed and tested this method to gradually guarantee the highest possible level of security and a more accurate report on the execution of phishing sites.

Emails containing malicious links or attachments are used by the attacker in this study [16] to achieve a number of goals, including the theft of sensitive information and the recording of conversations. Recipients of these emails put themselves at risk of identity theft and financial hardship. To identify and stop the phishing fraud, researchers employed a mix of defunding and semantic analysis techniques. Additionally, the text, connections, photos, and other data on the site are reviewed for design verification, and a library of malicious scams is created. At last, we compared our suggested setup to other methods already in use [16] and rated it

accordingly. The results validate the efficacy of our suggested approach in countering phishing attempts.

3. Methods

The analysis's individual modules are shown in Fig. 3's framework.

3.1. Dataset

We have included a malware dataset from public sources like Kaggle and custom data sets in our model. Our model is tested on 20% of the Kaggle phishing dataset and trained on 80% of the dataset. There are 95911 rows and 12 columns in the dataset, which includes information from both malicious and reputable websites [9].

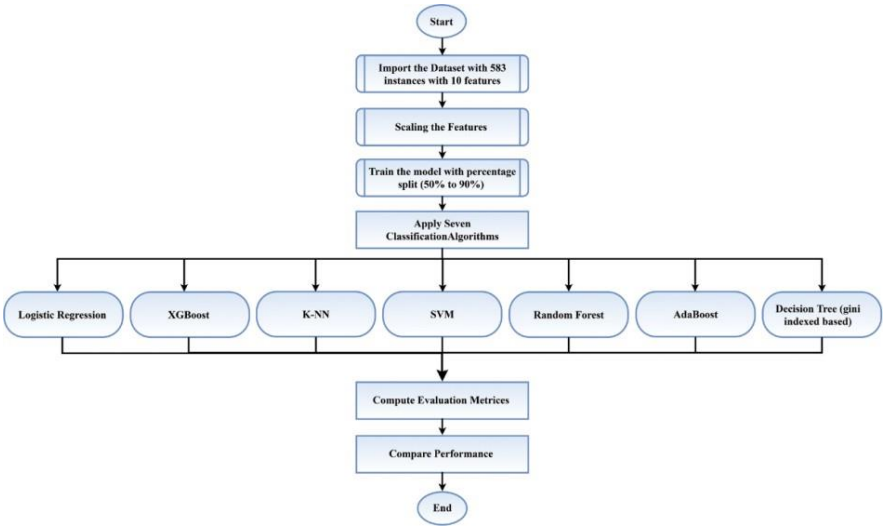

Fig. 3. Three types of machine learning.

3.2. Data pre-processing

Preprocessing data includes things like cleaning, occurrence identification, extraction of features, normalization, transformation, etc. The ultimate training dataset is the end result of data preparation. Interpreting final processing results may be affected by how the data was pre-processed. Steps including completing missing data, reducing noise, identifying, and discarding outliers, and settling compatibility issues are all possible within the context of data cleaning. The term "data integration" may refer to the process of combining many data sources. Collecting and standardizing data is what we call "data transformation," and it's what we use to get an accurate read on anything. Through data reduction, we may get a bird's-eye perspective of the information that is much more manageable in size but still contributes to the same analytical result [10].

3.3. Analysis of uncertain data

Data exploration (DAE) is a multi-method data analysis approach, as depicted in Fig. 4. DAE relies heavily on diagrams. It allows for the best possible understanding of a dataset by revealing its inner workings, extracting its most crucial parameters,

identifying its outliers and anomalies, and putting previously untested assumptions to the test. The Heat map is shown in Fig. 5.

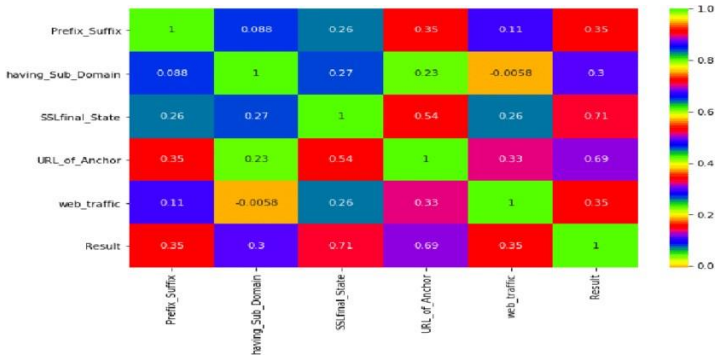

Fig. 4. Data exploration.

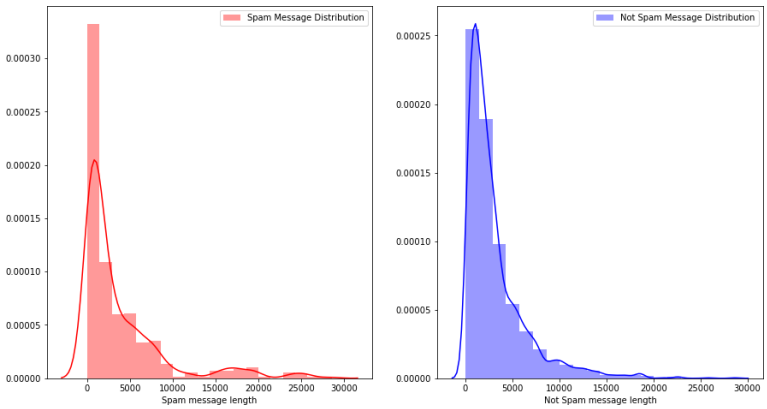

Fig. 5. Heatmap.

3.4. Train and test data split

The dataset is split into a training set and a testing set so that algorithms may be trained on the training set and then used to the testing set to identify phishing websites. In order for the modelling approach may efficiently train and learn from the data, the testing set is examined for 30% of the data.

3.5. Machine learning algorithms

3.5.1. Logistic regression

The likelihood of a binary answer may be predicted using logistic regression, a mathematical method. To predict a binary outcome (0 or 1), such as success or failure (yes or no), logistic regression is employed with a set of predictor variables. Logistic regression, like other types of regression models, is used for making predictions. It is used to visualize information and highlight the connection between a single binary dependent variable and several nominal, ordinal, interval, or ratio-level independent variables. A more intricate cost function is also required. In lieu of a linear function,

this price is expressed as a sigmoid function or logistic function. As demonstrated in Eq. (1), this algorithm's hypothesis favours the limit of the cost function between 0 and 1.

$$0 \leq h_o(x) \leq 1 \, (1) \tag{1}$$

3.5.2. K-nearest neighbours

Closest Neighbour Letter K. The simplest approach is the k-nearest neighbours (KNN) algorithm. It may be used to solve classification and regression issues. It's a common component in things like picture recognition software, simple recommendation engines, and automated decision-making platforms. KNN is used by major online retailers and streaming services like Amazon and Netflix to recommend a wide range of books to their customers. KNN is effective because it uses standard mathematical methods. To begin using KNN, it is necessary to transform data into their correct values. By emphasizing the gap between the numerical rates of these locations, we can see how it functions. As indicated in Eq. (2), the most common way to calculate such a distance is by utilizing the Euclidean distance.

$$(p,q) = (q,p) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2} \tag{2}$$

$$(p,q) = (p,q) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

KNN uses the above algorithm to calculate an approximate Euclidean distance and the test dataset. Then, it determines which combinations of points have the highest likelihood, depending on whether or not they include points that are similar to the test data. KNN is used for classification in order to determine which class has the highest frequency among the K most similar examples. The standardized prevalence of cases contributing to each class in the collection of K closest similar cases is used to calculate class probabilities for every new data occurrence. 0 and 1 are the only possible values for a category in a binary categorization.

p(class = 0) = count (class = 0) / (count(class = 0) + count (class = 1))

You can always break a bond by raising K by 1, at which point you look at the set of railway cars with the most similar cars together.

3.5.3. Decision tree

In a decision tree, an internal node represents a parameter, a branch represents a selection command, and a leaf node represents the result. The "root" node is the highest level in a decision tree. The parameter determines how the data is split up. Partitioning the tree in this way is called recursive partitioning, and it's rather clever. If you need assistance deciding, go to Fig. 6. Its human-like reasoning is reflected in its flowchart-like design. Because of this, decision trees may be better explained and understood. White-box algorithms are a specific sort of machine learning software. It's a description of the thought processes that underlie rational choice. Especially contrasted to a neural network-based approach, its training time is far shorter. The temporal complexity of the decision tree is proportional to the quantity of records and the number of parameters in the sample dataset. Because it does not reliant on

inference from probability distributions, the decision tree is considered a disburse as well as non-parametric technique. They can reliably process data in high dimensions. In addition to being able to spot non-linear patterns with relative ease, it also needs little data preprocessing—no need to standardize columns, for example. Feature engineering uses it too, for things like missing data detection.

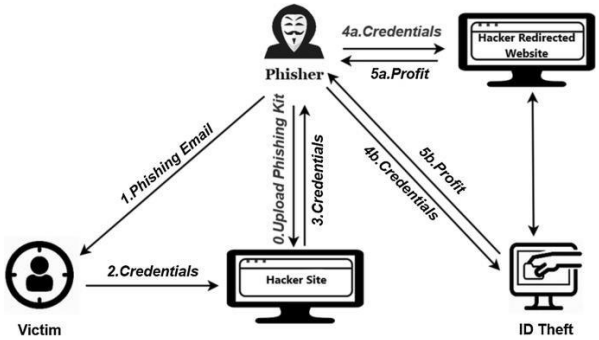

Fig. 6. Decision tree architecture.

3.5.4. Random forest

The supervised machine learning method known as "Random Forest," which involves combining many decision trees into one "forest," is widely used for both classification and regression tasks. It backed the concept of ensemble learning, which can be thought of as a method of combining several classifiers to solve a complex problem and improve the model's performance. With Random Forest, predictions are improved by combining numerous decision trees. The premise behind Random Forest is that using numerous models in combination might lead to considerably better results than using only one. Every tree in a Random Forest network casts a vote for the classification method's output when the network is used for that purpose. The most popular categorization is chosen by the forest. However, when using Random Forest for extrapolation, the forest incorporates information from all of the trees. It requires less time to train than other methods and produces consistent results within proper guidelines despite the fact that individual decision trees sometimes produce mistakes. You may trust its prediction of the outcome. It performs well, even when dealing with massive datasets. Additionally, its precision is maintained even in the absence of a very large data set. Bootstrap generates test samples for each algorithm using row sampling. With the use of aggregation, these small datasets may be reduced to a few key statistics for analysis and integration. Variance is an error that arises due to inconsistencies in the training dataset. As a consequence, high variance leads to the training of irrelevant or ambiguous information in the dataset, rather than the desired outcomes (the "signal"). Overfitting describes this problem's nature. While an over fitted model may do well in training, it will fail to differentiate between background noise and actual signal when put to the test. The bootstrap method incorporates the technology of bagging to account for large variations. In general, Random Forest works well, produces good results, and is developed really quickly. This simplified version of the Random Forest is seen in Fig. 7.

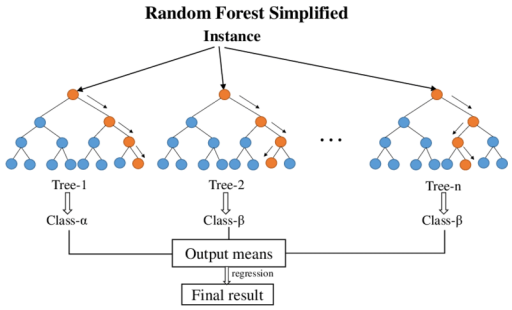

Fig. 7. Random forest architecture.

XG Boost. The acronym XG Boost refers to the process of eXtreme Gradient Boosting. Designed for efficiency and effectiveness, this method makes use of gradient enhanced decision trees. Boosting is a collaborative learning methodology that employs cutting-edge methods to correct flaws in previously presented models. Each new model is added in order until we reach a point where we can make no more improvements. In order to reduce the loss as it incorporates new models, it employs a gradient descent method. Using this technique effectively conserves resources like CPU time and RAM. The goal of the layout was to maximize the effectiveness of the available data for model training. The primary benefits of using XG Boost are its fast execution and high model performance. Regression and categorization models are both feasible using this method.

3.5.5. AdaBoost

For Adaptive Boosting, you may just call it AdaBoost. It's a method for learning how to use machine learning. The AdaBoost model has the major benefit of being quick, easy, and straightforward to implement. The technique is adaptable enough to work with a wide variety of machine learning frameworks. In order to solve the classification problem, it might turn ineffective learners or predictors into effective ones.

3.6. Deep learning algorithms

Artificial Intelligence. Neural Network. Inspired by the human nervous system, Artificial Neural Networks (ANNs) may learn from data by seeing real-world scenarios and forming inferences accordingly. In order to test a hypothesis, ANN can establish a correlation between exogenous and endogenous factors. Complicated insights and hidden meanings are mined from the data collection. With no assumptions regarding the statistical representation of the aspect, the connection between independent variables and dependent variables may be constructed. It improves regression algorithms in useful ways, for example by making them more capable of dealing with noisy data. Input nodes and output nodes are both part of an ANN's hidden node structure. The evidence is sent from the input nodes layer to the hidden nodes layer, where the information is either activated or left dormant depending on the stimulation functions sent. The weighted functions are used as proofs in the hidden layers, and whenever the value of a node or device on the network in this layer reaches a threshold, the value is passed on to one or more nodes in the output layer. There must be a large number of datasets used to train the ANN model. When there is insufficient data to train an ANN, such as in the case of rare or extreme occurrences, they cannot be used. Artificial neural networks do not allow the manifestation of creative intelligence to stand in for demonstrable expertise.

4. Results and Discussion

The work is carried out in an environment called Anaconda, and the dataset utilized in this study was obtained from the "Kaggle" website. To ensure optimal model performance, this information is first thoroughly inspected for any missing, duplicate, or otherwise inaccurate data. The dataset is explored by using the matplot and seaborn modules. Label Encoder is used to convert string arguments to an integer data type. A 70:30 split is then made between the training and testing datasets, for example. Python's sklearn package is used to create classifier models, which are then tested, trained, and evaluated based on the collected data. In order to implement ensemble algorithms, the module ensemble is used. Also, each model is put through its paces of testing and training. The keras module in TensorFlow is used to train and evaluate a deep-learning algorithm. At the end, we use the chart-studio module to visualize the data we've collected and compare the accuracy of each method we've considered. Several classifications and ensemble techniques have been used to train and test the spoofing webpage detection model, allowing for a thorough comparison of the model's performance. After all algorithms have returned their results, each will provide an evaluation of how well it performed. Table 1 displays the results of these comparisons between the various methods. The accuracy of each method is shown in the binary classification for easy comparison. A deep learning technique is also used to train the dataset. Figures 8 and 9 show the final results of the algorithms' accuracy comparison.

Table 1. Trading and testing accuracy for all algorithms.

S. No.	Algorithm	Training set Accuracy	Testing set Accuracy	Precision Score
1	Logistic Regression	78.00	78.00	80.27
2	Ada Boost	85.00	85.92	85.72
3	XG Boost	92.80	92.48	93.00
4	Random Forest	96.00	95.0	94.10
5	K Nearest Neighbour	96.75	93.11	93.84
6	Decision Tree	100	95.51	96.14

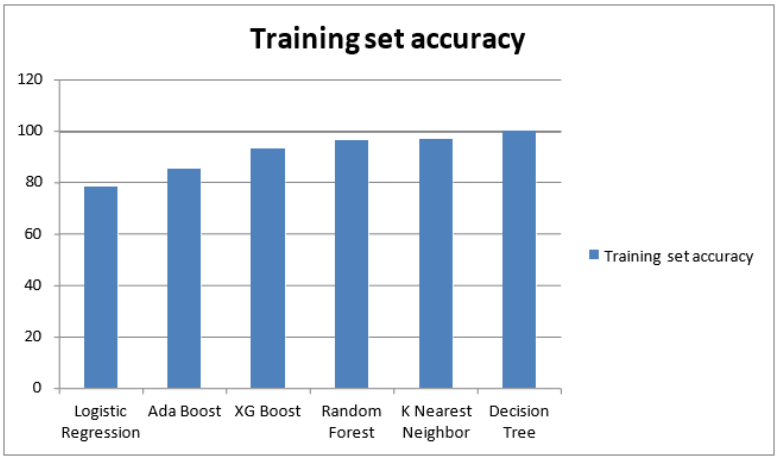

Fig. 8. Training set accuracy.

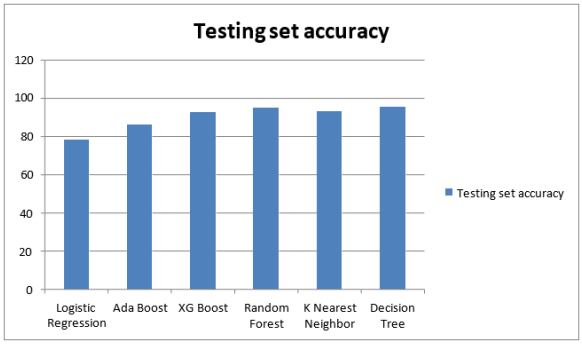

Fig. 9. Testing set accuracy.

5. Conclusions

Based on the findings, it is clear that perhaps the Model of Decision Trees is the most effective and precise option. However, ensemble methods have also been shown to be useful because to their speed, performance, and ability to utilize several classifiers to improve prediction accuracy. Phishing attacks on websites are increasingly dangerous now. It poses an increasingly serious risk to people's everyday activities and online communities. In these intrusions, the attacker poses as a reputable business in order to steal sensitive data. Our approach provides a potent tool for identifying phished domains, which bodes well for future efforts to defend against these types of assaults.

Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Networks
BPNN	Back Propagation Neural Network
CSV	Comma Separated Values
DAE	Data Exploration
DT	Decision Tree
KNN	K Nearest Neighbour
LR	Logistic Regression
ML	Machine Learning
RF	Random Forest
XG Boost	Extreme Gradient Boost

References

1. Das, A; Baki, S.; El Aassal, A.; Verma, R.; and Dubar, A. (2019). SoK: A Comprehensive reexamination of phishing research from the security perspective. *IEEE Communications Surveys & Tutorials*, 22(1), 671-708.
2. Wu, J.; Yuan, Q.; Lin, D.; You, W.; Chen, W.; Chen, C.; and Zheng, Z. (2022). Who are the phishers? Phishing scam detection on Ethereum via network embedding. *IEEE Transactions on Systems, man, and Cybernetics: Systems*, 52(2), 1156-1166.
3. Chen, S.; Fan, L.; Chen, C.; Xue, M.; Liu, Y.; and Xu, L. (2019). GUI-squatting attack: Automated generation of android phishing apps. *IEEE Transactions on Dependable and Secure Computing*, 18(6), 2551-2568.

4. Li, Q.; Cheng, M.; Wang, J.; and Sun, B. (2020). LSTM based phishing detection for big email data. *IEEE Transactions on Big Data*, 8(1), 278-288.
5. Allodi, L.; Chotza, T.; Panina, E.; and Zannone, N. (2020). The need for new antiphishing measures against spear-phishing attacks. *IEEE Security & Privacy*, 18(2), 23-34.
6. Niu, X.; Liu, G.; and Yang, Q. (2020). OpinionRank: Trustworthy website detection using three valued subjective logic. *IEEE Transactions on Big Data*, 8(3), 855-866.
7. Devikanniga, D.; Ramu, A.; and Haldorai, A. (2020). Efficient diagnosis of liver disease using support vector machine optimized with crows search algorithm. *EAI Endorsed Transactions on Energy Web*, 7(29), e10.
8. Anandakumar, H.; and Umamaheswari, K. (2017). Supervised machine learning techniques in cognitive radio networks during cooperative spectrum handovers. *Cluster Computing*, 20(2), 1505-1515.
9. Zhu, E.; Liu, D.; Ye, C.; Liu, F.; Li, X.; and Sun, H. (2018). Effective phishing website detection based on improved bp neural network and dual feature evaluation. *Proceedings of the* 2018 *IEEE Intl Conf on Parallel & Distributed Processing with Applications*, *Ubiquitous Computing & Communications*, *Big Data & Cloud Computing*, *Social Computing & Networking*, *Sustainable Computing & Communications* (*ISPA/IUCC/BDCloud/SocialCom/Sustain Com*), Melbourne, VIC, Australia, 759-765.
10. Garcés, I.O.; Cazares, M.F.; and Andrade, R.O. (2019). Detection of phishing attacks with machine learning techniques in cognitive security architecture. *Proceedings of the* 2019 *International Conference on Computational Science and Computational Intelligence* (*CSCI*), Las Vegas, NV, USA, 366-370.
11. Zekri, M.; El Kafhali, S.; Aboutabit, N.; and Saadi, Y. (2017). DDoS attack detection using machine learning techniques in cloud computing environments. *Proceedings of the* 2017 3*rd International Conference of Cloud Computing Technologies and Applications* (*CloudTech*), Rabat, Morocco, 1-7.
12. Touqeer, H.; Zaman, S.; Amin, R.; Hussain, M.; Al-Turjman, F.; and Bilal, M. (2021). Smart home security: challenges, issues, and solutions at different IoT layers. *The Journal of Supercomputing*, 77(12), 14053-14089 .
13. Mohith Gowda, H.R.; Adithya, M.V.; Gunesh Prasad, S.; and Vinay, S. (2020). Development of anti-phishing browser based on random forest and rule of extraction framework. *Cybersecurity*, 3:20, 1-14.
14. Korkmaz, M.; Sahingoz, O.K.; and Diri, B. (2020). Detection of phishing websites by using machine learning-based URL analysis. *Proceedings of the* 2020 11*th International Conference on Computing*, *Communication and Networking Technologies* (*ICCCNT*), Kharagpur, India, 1-7.
15. Guillod, T.; Papamanolis, P.; and Kolar, J.W. (2020). Artificial neural network (ANN) based fast and accurate induction modeling and design. *IEEE Open Journal of Power Electronics*, 20(1), 284-299.
16. Zhang, S.; Li, X.; Zong, M.; Zhu, X.; and Wang, R. (2018). Efficient kNN classification with different numbers of nearest neighbors. *IEEE Transactions on Neural Networks and Learning Systems*, 29(5), 1774-1785 .