Research Reports on Computer Science
https://ojs.wiserpub.com/index.php/RRCS/

UNIVERSAL WISER
PUBLISHER

Research Article in Special Issue: Selected Papers from the 4th International Conference on Machine Learning, Image Processing, Network Security and Data Sciences (MIND-2022)

Machine Learning and Deep Learning for Phishing Page Detection

Swatej Patil[1*], **Mayur Patil**[2], **Kotadi Chinnaiah**[1]

[1]Department of Computer Science and Engineering, G. H. Raisoni College of Engineering, Nagpur, India
[2]Department of Computer Science, Rashtrasant Tukadoji Maharaj Nagpur University (RTMNU), Nagpur, India
E-mail: swatejpatil007@gmail.com

Received: 8 March 2023; **Accepted:** 28 March 2023

Abstract: The term "phishing" is often used to describe an attempt to obtain confidential data such as passwords or credit card details by impersonating a trustworthy source. In most cases, the term refers to attempts to trick users into providing sensitive information in response to a fraudulent email or web page. However, the term is also used to describe a broader category of online attacks to obtain sensitive information or to disrupt services or systems. Incorporating different machine learning and deep learning algorithms, including Support Vector Machine (SVM), Gradient Boosting Machine (GBM), and random forest, the authors of this research presented a technique for identifying phishing websites. The data sets from PhishTank and the University of New Brunswick were used to train and test the learning models. The XGboost model was able to surpass most existing techniques by achieving a maximum accuracy of 86.8%. This technique can be used in modern web browsers like Google Chrome and Mozilla Firefox to accurately detect phishing websites.

***Keywords*:** phishing, cybersecurity, machine learning, deep learning

1. Introduction

The term "phishing" is often used to describe an attempt to obtain confidential data such as passwords or credit card details by impersonating a trustworthy source. In most cases, the term refers to attempts to trick users into providing sensitive information in response to a fraudulent email or web page. However, the term is also used to describe a broader category of online attacks to obtain sensitive information or to disrupt services or systems. The most common types of phishing attacks exploit the existing trust between users and companies to trick users into providing sensitive information.

The most common form of phishing involves sending an email that appears to be from a legitimate source, like a bank or other financial institution, but which actually contains a link or disguised link that will download malware onto the victim's computer if they follow the link or are redirected to the malware. Some phishing emails will appear to be emails that appear to be from the company the victim is trying to contact, but which actually contain phishing pages or malware. While most of us have received phishing emails at some point in our lives, many don't realize that they are still occurring today. The goal of a phishing email is to trick you into providing your information. Often, the text of a

DOI: https://doi.org/10.37256/rrcs.2320232629

phishing email will mimic the language of a legitimate company or organization.

According to the study [1], women have a higher chance of being the victims of phishing attacks than men. Furthermore, individuals in the age group of 18 to 25 are more prone to phishing attacks than any other age group. Why are phishing attacks so harmful to normal people? And why are they such a problem for younger individuals? The answer to both of those questions lies in the nature of the social and technical infrastructure that we rely on for modern life. Most people experience a phishing attack only once or twice in their lives. Most people don't think much about when they receive a phishing email. The messages appear harmless enough, and most people click through to the requested page or site without realizing they've been tricked. However, phishing attacks have a much broader impact than most people realize. Many people may not realize the extent to which phishing attacks are a threat to their personal and financial security, and the harm they can cause to the security of the internet as a whole. Phishing attacks are one of the most prevalent ways for cybercriminals to gain personal information and account information from victims. They are also one of the most common ways in which malicious hackers gain access to victims' accounts. Because of their prevalence, phishing attacks have become a popular method for hackers to target large numbers of victims. However, although they have become a common occurrence, they are still capable of causing a great deal of damage.

Approximately two million phishing websites have been identified as legitimate; 11,000 of them are still accessible online, while nearly two million more have been taken offline [2]. These numbers represent only a small fraction of the total number of phishing websites, which are thought to number in the hundreds of millions. The vast majority of phishing websites are detected and taken down by security professionals, but a small number of sites continue to evade detection. This evolution of phishing has led to more sophisticated and convincing content, which has made it even harder for users to detect and avoid being tricked. They use valid domain names, convincing domain front pages, and legitimate-looking emails to lure in potential victims.

The manuscript is structured so that the first section goes over the background and significance of phishing page detection. A summary of well-known studies on the subject of phishing page detection is given in the second part, along with prospects for further study. The data set source and its attributes are briefly described in the next section. In the fourth part, examples of common machine learning implementations are shown. In the conclusion section, a summary of the study and potential prospects for phishing page identification research are given.

2. Literature review

This section gives a synopsis of previous research on phishing pages and their detection. The authors of [2] have concentrated on increasing the performance of the phishing page detection model. To improve the efficiency and accuracy of the detection model, the authors suggested a rule-based method. Numerous machine learning methods have been examined in an effort to improve detection accuracy. To identify phishing pages, the authors created an algorithm for choosing the most relevant parameters, such as the Hypertext Transfer Protocol (HTTP), Uniform Resource Locator (URL) length, the amount of dots (.) in the URL, the favicon, free hosting or inexpensive domain names, and Hypertext Markup Language (HTML) file size. They utilized a data set that included 500 authentic websites and 500 fraudulent websites. The authors accurately recognized 86.6% of the phishing pages, whereas just 13.4% of them were not.

The primary objective of [3] was to use deep learning techniques to identify phishing websites using URLs. They have created a real-time detection method using deep neural networks. The Ebbu2017 data set was utilized in the creation of a deep learning model. There were 73,575 total URLs in the sample, of which 37,175 were phishing URLs and the remainder were real URLs. One significant disadvantage of this heuristic method is that it might occasionally identify legitimate URLs as phishing URLs.

The authors of [4] developed a unique method to identify phishing websites by capturing every hyperlink on a web page using an application programming interface (API) from Google and building a parse tree out of the retrieved hyperlinks. This model may also determine whom the phishing web pages are targeting with the use of a parse tree validation mechanism. The authors utilized 1,000 genuine websites and 1,000 phishing websites for the development and testing of the model. Among the 2,000 web pages, 123 phishing pages are wrongly classified as genuine, while 143 valid pages are wrongly classified as phishing. The authors enhanced the model's performance by including these incorrectly detected web pages as input. They attained a false positive rate of 5.20%, while the achieved false negative rate was 7.30%.

Sanglerdsinlapachai et al. [5] suggested a method for phishing domain identification in which they utilized an extra domain top-page similarity characteristic and integrated Carnegie Mellon Anti-phishing and Network Analysis Tool (CANTINA). The error rate of their approach was 7.50%. In this method, learning algorithms including Naive Bayes, Neural Network, Support Vector Machine, Random Forest, J48 Decision Tree, and AdaBoost are employed. However, the data set utilized for the initial analysis is somewhat limited, with just 200 URLs, 100 of which were phishing URLs.

In [6], the authors developed a data mining-based technique for phishing URL detection. Their MCAC approach, which stands for Multi-label Classifier-based Associative Classification, works in three separate stages. The method iterates through the training data in the first stage to find the distinctive and significant characteristics. In order to establish the classification directive, the rules are ordered in step two according to reliability, duration, and support. In the third and final stage, the rules are used to classify the URLs with greater support and reliability. Their approach was able to achieve an accuracy of over 90%.

3. Overview of the data set

The final data set was created using the concatenation of two data sets of genuine as well as phishing websites' URLs as mentioned in Table 1 and Figure 1. The phishing URLs were gathered from a well-known data set source, PhishTank [7]. 5,000 phishing URLs were randomly taken from PhishTank and the shape of the data set was 5000 × 8. The genuine URLs were obtained from a data set at the University of New Brunswick [8]. Here, we also generated 5,000 random URLs. The data sets are then used for feature extraction.

Table 1. Description of database

URL type	No. URLs	Features	Source
Genuine	5,000	1	University of New Brunswick
Phishing	5,000	8	PhishTank

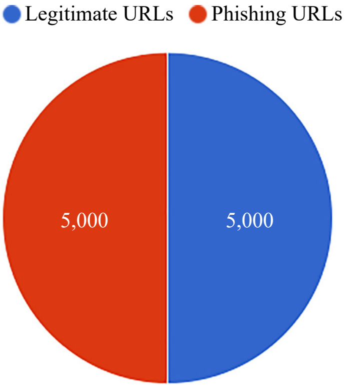

Figure 1. Legitimate and phishing URL counts

4. Implementation and methodology

4.1 *Features extraction*

After gathering the genuine and phishing data sets, feature extraction is the initial stage in our research process. After that, the features are categorized into three groups. The first of them is concerned with the URLs or address bar of a web page, including the length, domain, and number of dots (.) present in the URL. The Domain Name System (DNS) record of the web page, traffic age of the URL, and end period of the URL fall under the second group of domain-based characteristics. The last category focuses on a web page's HTML/ Cascading Style Sheets (CSS)- and JavaScript-based features.

After the feature extraction was finished, the two data sets were then concatenated to produce a single data set consisting of 10,000 rows and 16 feature columns. Then, machine learning models are trained and tested using this data set, shown in Figure 2. The features of our data set are as follows:

1. **Domain of URL:** The domain of the URL doesn't provide any useful clues as to whether the provided link is a phishing attempt or not.
2. **Internet Protocol (IP) address in URL:** A URL that has an IP address rather than a domain name may be a clear sign that it is a phishing URL.
3. **"@" symbol in URL:** The @ sign in the URL indicates that anything preceding the symbol is disregarded by the browser, and the real URL is found after the symbol.
4. **Length of URL:** There is no specific length of URL that can indicate whether the given URL is phishing or not, however, the phishers may use a long URL to hide the suspicious part of a URL.
5. **Depth of URL:** A URL's depth may be determined by counting the number of subpages it contains. To make the user think he or she is dealing with a legitimate URL, the phishers may add a subdomain to the URL.
6. **Redirection "//" in URL:** The inclusion of "//" in the URL indicates that the user is being redirected to another website.
7. **"HTTP/HTTPS" in the domain name:** To trick the user, the phisher might include a bogus HTTP Secure (HTTPS) token in the URL.
8. **Using URL shortening services "TinyURL":** Using a URL shortening service may significantly reduce the length of a URL, which can effectively disguise a suspicious URL.
9. **Prefix or Suffix "-" in Domain:** The phishers employ a number of strategies to alter a suspected phishing URL so that it resembles a legitimate URL. A gullible person might be duped by a genuine URL that uses a prefix or suffix.
10. **DNS record:** The URL may be a phishing site if the DNS record is missing or nonexistent.
11. **Website traffic:** Real websites receive a lot of everyday traffic from users. Phishing websites, on the other hand, have a brief existence and hence receive very little online traffic.
12. **Age of domain:** Websites with fewer than 6 months of internet presence are considered unsafe.
13. **End period of domain:** The expiration date of a domain may be estimated using the WHOIS database. If the expiration date of a domain is fewer than six months, it may be regarded as suspicious.
14. **IFrame redirection:** The HTML element IFrame is used to include another web page into the one that is now being shown. The phishers may include a phishing web page in a legitimate page.
15. **Status bar customization:** The phisher can display a bogus URL in the status bar by using JavaScript.
16. **Disabling right-click:** JavaScript can be used by the phisher to block the right-click, preventing users from viewing the website's source code.

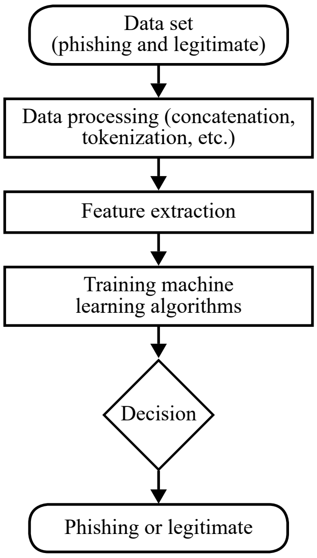
Data set
(phishing and legitimate)
Data processing (concatenation, tokenization, etc.)
Feature extraction
Training machine learning algorithms
Decision
Phishing or legitimate

Figure 2. Flow diagram of study

5. Training the machine learning models

The data set has to be preprocessed before the machine learning models can be trained and assessed. Additionally, the data set's initial 5,000 rows contain real URLs, whereas the latter 5,000 rows contain phishing URLs. The data set is then shuffled to create a more objective detection model. Following that, the data set is divided into training and testing data sets. 80% of the data from the original data set are included in the training data set, while the remaining 20% are included in the testing data set.

5.1 *XGboost*

XGBoost was designed primarily for increased computation speed and enhanced performance by leveraging gradient-boosted decision trees. Extreme Gradient Boosting, often known as XGBoost, aids in incorporating the most of hardware and memory resources for tree boosting algorithms. It may be used in computer settings and offers the advantages of algorithm improvement and model tuning. XGBoost supports the three primary gradient boosting methods: gradient boosting, normalized boosting, and stochastic boosting. It differs from other libraries because it also enables the addition and fine-tuning of regularization parameters. The technique makes the best use of memory resources while being very successful at cutting down on calculation time [9]. In our model, the XGboost algorithm was able to attain an accuracy of 86.8%, as shown in Figure 3.

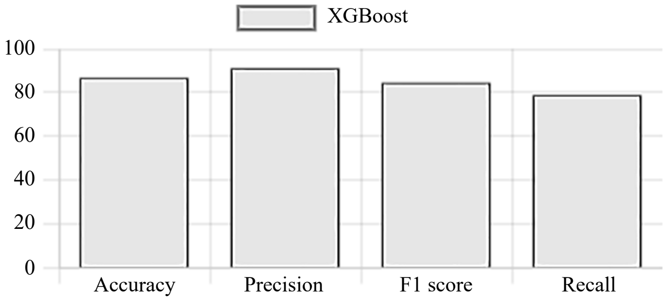

Figure 3. Performance parameters for XGBoost

5.2 *Decision tree*

Decision trees are algorithms that group occurrences according to the values of their features. A decision tree's nodes represent features in instances that are waiting to be categorized. To divide the training data, a feature is chosen, and that feature becomes the root node. Following a similar process, the tree begins to branch out and create sub-trees until the same class subsets are established [9]. The decision tree algorithm managed to achieve an accuracy of 81.3% in our model, as shown in Figure 4.

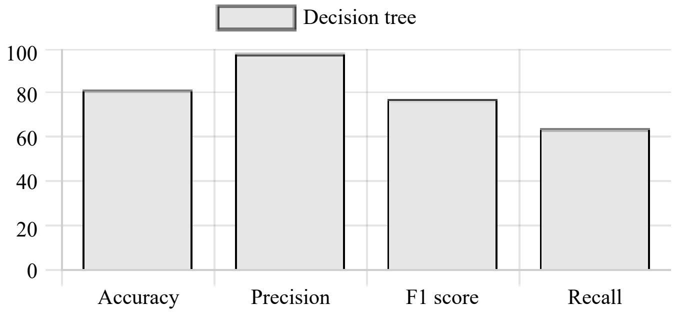

Figure 4. Performance parameters for decision tree

5.3 *Multilayer perceptrons*

In place of more conventional statistical methods, multilayer perceptrons can be a useful substitute. When given fresh, untested data, it may be trained to generalize with accuracy and represent extremely nonlinear functions. Multilayer perceptrons don't get into the data distribution with any preconceived notions. Multilayer perceptrons are supervised in their learning. An output layer comes last in a multilayer perceptron, which may also include one or more hidden layers. Each node in multilayer perceptrons is said to be completely coupled to every other node in the layer above and below [10]. In our model, the multilayer perceptron algorithm was able to attain an accuracy of 86.6% with 93.2% precision, as shown in Figure 5.

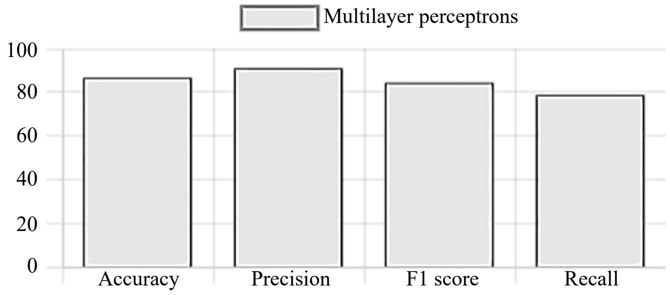

Figure 5. Performance parameters for multilayer perceptrons

5.4 *K-Nearest Neighbor (KNN)*

A case-based learning approach called KNN retains all of the training data for classification. Due to its sluggish learning nature, it is not suitable for many applications, including dynamic web mining for big repositories. Finding some representatives to represent the entire training data set for classification, i.e., using the training data set to create an intuitive learning model and using the resulting model for classification, is one technique to increase its efficiency. Numerous current techniques, like decision trees and neural networks, were first created to construct such a model. The performance of various algorithms is one of the evaluating criteria [11]. The KNN algorithm was able to achieve an accuracy of 82.8% in our model.

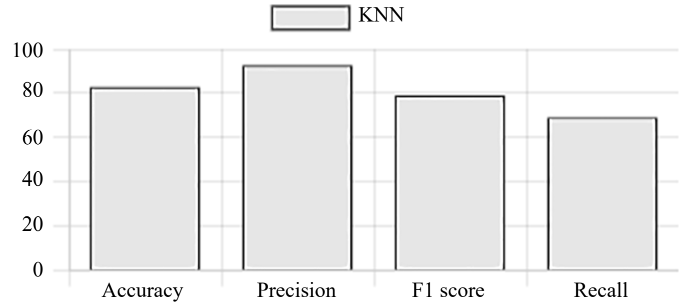

Figure 6. Performance parameters for KNN

5.5 *Random forest classifier*

The bagging technique is extended by the random forest algorithm, which uses both feature randomness and bagging to generate an uncorrelated forest of decision trees. Each decision tree in the ensemble that makes up the random forest method is built of a data sample taken from a training set with a replacement known as the bootstrap sample. A random forest and decision tree vary in that a decision tree takes into account all possible feature splits whereas a random forest merely chooses a portion of those features. There are three key hyperparameters for random forest algorithms that must be specified prior to training. Node size, tree count, and sampled feature count are a few of them [12]. The random forest algorithm managed to achieve an accuracy of 81.9% in our model with 97.9% precision and 0.636 recall.

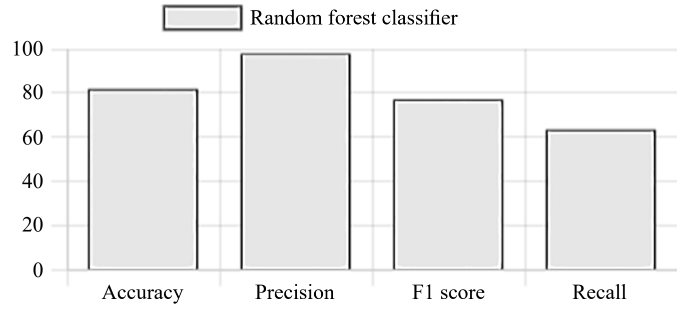

Figure 7. Performance parameters for random forest classifier

5.6 *Gradient Boosting Machine (GBM)*

The AdaBoost algorithm should first be discussed in order to better understand the gradient boosting algorithm (GBA). The AdaBoost algorithm starts by training a decision tree with equal weights for each observation. Gradient tree boosting, gradient-enhanced regression trees, and Gradient Boosted Regression Trees (GBRT) are other names for GBM. It is intended to improve the performance of classification and regression trees, a type of classifier that does both categorization and regression at once. A GBM is a component of a homogeneous ensemble, in which a prediction model is created by combining multiple weak classifiers of the same kind (weak prediction models). Multiple models are trained gradually, additively, and sequentially using gradient boosting [13]. In our model, the GBM algorithm was able to attain an accuracy of 81.3%.

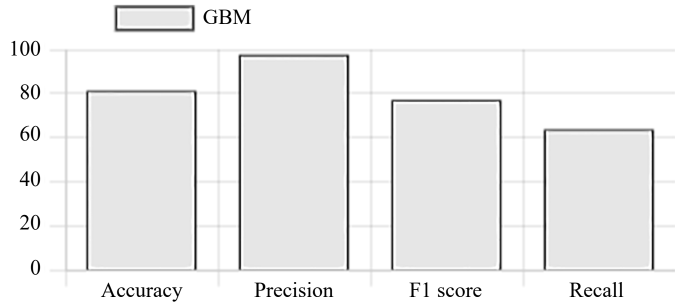

Figure 8. Performance parameters for GBM

5.7 *Support Vector Machine (SVM)*

SVMs include a class of supervised learning methods for data classification, regression analysis, and outlier detection. SVMs have a great deal of potential in high-dimensional spaces. It is also quite effective in managing memory. Drawing a straight line between two classes is how a straightforward linear SVM classifier functions. Accordingly, the data directed on a single side of the line will all be allocated to one category, while the data points on the different sides of the line will be assigned to a different category. This implies that the number of possible lines is unlimited. Avoid over-fitting when selecting kernel functions and regularization terms, though, if the number of feature traits is significantly higher than the number of data samples [14]. The SVM algorithm managed to achieve an accuracy of 80.1% in our model.

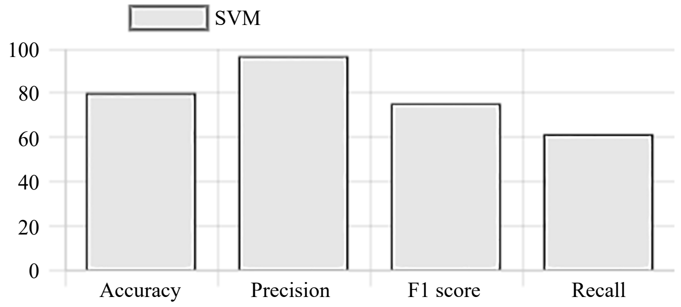

Figure 9. Performance parameters for SVM

6. Results and discussions

After thorough testing of the models using a variety of legitimate and phishing URLs, XGBoost has the greatest detection accuracy of all the models (86.0%), followed by multilayer perceptrons (85.8%). The SVM's accuracy was the lowest at 80.1%.

Table 2. Results of different machine learning algorithms

Serial no.	Algorithms	Training accuracy	Testing accuracy	Recall	F1 score	Precision	Support
1	XGBoost	86.8%	86.0%	0.789	0.846	0.911	977
2	Multilayer perceptrons	86.6%	85.8%	0.766	0.84	0.932	977
3	KNN	82.8%	82.2%	0.691	0.791	0.925	977
4	Random forest	81.9%	81.6%	0.636	0.771	0.979	977
5	Decision tree	81.3%	81.5%	0.638	0.771	0.975	977
6	GBM	81.3%	81.5%	0.638	0.771	0.975	977
7	SVM	80.1%	80.3%	0.617	0.754	0.968	977

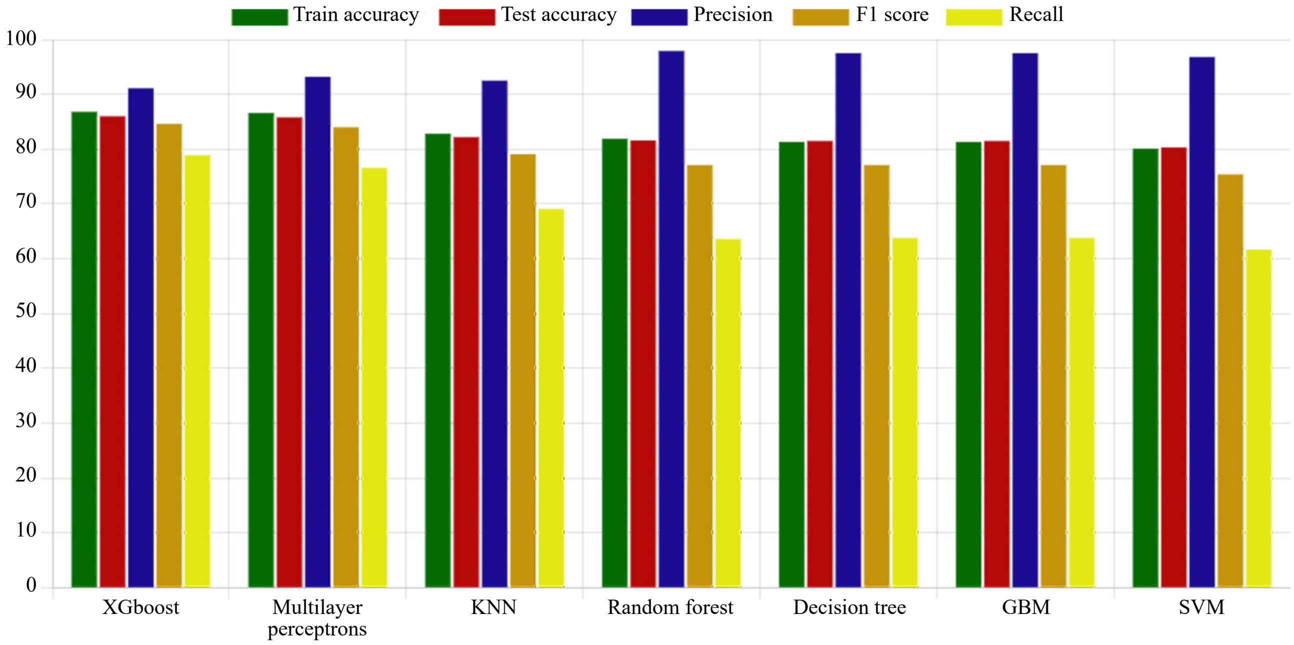

Figure 10. Performance parameters comparison of different machine learning algorithms

7. Conclusion and future scope

The first portion of the article offered an overview of well-known studies on phishing page detection. The classification and detection of phishing page URLs using machine learning and deep learning has been the main emphasis of this paper. The data sets for training and testing were acquired from PhishTank and the University of New Brunswick, respectively. The findings indicate that this method may accurately detect phishing URLs with a rate of 87.2%. The accuracy of the suggested method can be improved in the future by making use of a sizable data set and a number of additional features of websites and URLs. This technique may also be used to create a browser plugin that detects phishing URLs.

Conflict of interest

The authors declare that there is no conflict of interest.

References

[1] Sheng S, Holbrook M, Kumaraguru P, Cranor LF, Downs J. Who falls for phish? A demographic analysis of phishing susceptibility and effectiveness of interventions. In: *CHI'10: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, United States: Association for Computing Machinery; 2010. p.373-382. https://doi.org/10.1145/1753326.1753383

[2] Faris H, Yazid S. Phishing web page detection methods: URL and HTML features detection. In: *2020 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS)*. Bali, Indonesia: IEEE; 2021. p.167-171. https://doi.org/10.1109/iotais50849.2021.9359694

[3] Singh S, Singh MP, Pandey R. Phishing detection from URLs using deep learning approach. In: *2020 5th International Conference on Computing, Communication and Security (ICCCS)*. Patna, India: IEEE; 2020. p.1-4. https://doi.org/10.1109/icccs49678.2020.9277459

[4] Shyni CE, Sundar AD, Ebby GE. Phishing detection in websites using parse tree validation. In: *2018 Recent Advances on Engineering, Technology and Computational Sciences (RAETCS)*. Allahabad, India: IEEE; 2018. p.1-4. https://doi.org/10.1109/raetcs.2018.8443961

[5] Sanglerdsinlapachai N, Rungsawang A. Using domain top-page similarity feature in machine learning-based web phishing detection. In: *2010 Third International Conference on Knowledge Discovery and Data Mining.* Phuket, Thailand: IEEE; 2010. p.187-190. https://doi.org/10.1109/wkdd.2010.108

[6] Abdelhamid N, Ayesh A, Thabtah F. Phishing detection based associative classification data mining. *Expert Systems with Applications.* 2014; 41(13): 5948-5959. https://doi.org/10.1016/j.eswa.2014.03.019

[7] PhishTank. *Developer Information*. https://www.phishtank.com/developer_info.php [Accessed 15th February 2023].

[8] Mohammad RMA, McCluskey L, Thabtah F. (2015) *Phishing Websites Data Set*. UCI Machine Learning Repository. Data set. https://archive.ics.uci.edu/ml/datasets/Phishing+Websites

[9] Dhaliwal SS, Nahid AA, Abbas R. Effective intrusion detection system using XGBoost. *Information*. 2018; 9(7): 149. https://doi.org/10.3390/info9070149

[10] Gardner MW, Dorling SR. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric Environment*. 1998; 32(14-15): 2627-2636. https://doi.org/10.1016/s1352-2310(97)00447-0

[11] Guo G, Wang H, Bell D, Bi Y, Greer K. KNN model-based approach in classification. In: Meersman R, Tari Z, Schmidt DC. (eds.) *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE. OTM 2003*. Lecture Notes in Computer Science, vol 2888. Berlin, Germany: Springer; 2003. p.986-996. https://doi.org/10.1007/978-3-540-39964-3_62

[12] International Business Machines Corporation. *What Is Random Forest?* https://www.ibm.com/cloud/learn/random-forest [Accessed 15th February 2023].

[13] Tama BA, Rhee KH. An in-depth experimental study of anomaly detection using gradient boosted machine. *Neural Computing and Applications*. 2019; 31: 955-965. https://doi.org/10.1007/s00521-017-3128-z

[14] Scikit-learn developers. *1.4. Support Vector Machines*. https://scikit-learn.org/stable/modules/svm.html [Accessed 15th February 2023].