Journal of
Data Acquisition and Processing

ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.7766139

HAZARD IDENTIFICATION AND DETECTION USING MACHINE LEARNING

B S Panda[1], Dabbiru Chaturya[2], Injeti Gautham Sahil[3], Chalapaka Dinesh[4], Annam Bhanu Prakash[5]

[1]Professor, Dept of CSE, Raghu Engineering College, Visakhapatnam, India.
[2,3,4,5] Students, Department of CSE, Raghu Engineering College, Visakhapatnam
[1]panda.bs@raghuenggcollege.in, [2]19981a0536@raghuenggcollege.in,
[3]19981a0559@raghuenngcollege.in
[4]20985A0504@raghuenggcollege.in, 519981a0511@raghuenggcollege.in

Abstract

In the present day, web browsing has taken on significant importance in our daily lives. Nevertheless, this ease carries the risk of coming across malicious websites that could infect our devices with malware and steal our personal data. The present cybersecurity techniques, such firewalls and antivirus software, usually fall short of protecting us from these ever-evolving threats. As a result, a more sophisticated and useful model is needed that can accurately distinguish between safe and harmful online pages. This motivates us to develop a new clssification system that utilizes a range of machine learning classification algorithms, such as Adaboost, XGBoost, Random Forest, Support Vector Machine, Naive Bayes, and Logistic Regression, Decision Tree, K Nearest Neighbors, ANN, and Gradient Boosting in addition to analyzing and detecting URL-based features. Researchers have identified several machine learning classifiers, including Adaboost, XGBoost, and Support Vector Machine, that are effective in detecting malicious websites. In order to improve user online security, our aim is to develop a system that can accurately identify a web page's malicious intent. To achieve this, we will extract relevant attributes from web pages and train the classifiers using bagging and boosting methods. Our approach will be put to the test on a big dataset of web pages, and its efficacy will be compared to that of other approaches. Our results show how effectively our categorization algorithm can identify potentially harmful websites. Our proposed methodology can also significantly improve web security while protecting users from harm by providing a rapid and accurate way to discover and analyze hazardous websites. The findings of this study have wide-ranging implications that might have an impact.

Keywords: Malicious websites, Adaboost, XGBoost, Random Forest, SVM, Logistic Regression, ANN, Gradient Boosting.

I. INTRODUCTION

With more and more services becoming available online, such as e-commerce, online banking, social networking, and other things, using the internet has become a necessary part of our everyday life. The sophistication and advancement of internet browsers, however, also creates new opportunities for hackers to access user devices by taking advantage of flaws in software and hardware. This has made a quick and precise mechanism for identifying legitimate and fraudulent web pages necessary. Our study suggests a novel method for identifying malicious web sites based on extracted characteristics that makes use of a variety of a group of popular machine learning classifiers, including Adaboost, XGBoost, Random Forest, Support Vector Machine, Naive Bayes, and Logistic Regression, Decision Tree, K Nearest Neighbors, ANN, and Gradient Boosting and unique URL-based classifiers. The suggested system is trained using a combination of bagging and boosting strategies to improve accuracy and reduce false positives.

We emphasize the shortcomings of existing blacklisting services, which frequently contain inaccurate listings and fall behind the evolving methods employed by cyber criminals, in order to demonstrate the need for our suggested strategy. The goal of our research is to create a more accurate and efficient method of safeguarding people from potential harm through the exact detection[1] and avoidance of dangerous internet activity. Our research questions are focused on the viability of utilizing machine learning classifiers[2] to differentiate between benign and harmful online sites as well as the efficacy of the suggested system in identifying dangerous web pages in order to achieve this.

We have two things to offer the pitch: In order to improve web security and safeguard consumers from cyber attacks, we first offer a more accurate and effective technique to spot and stop dangerous online actions. Second, we present a fresh amalgamation of machine learning classifiers and methods that can serve as a model for further investigation in this field. The background, motivation, definitions, literature review, research gap, problem statement, purpose, remedy, contribution, and structure are all included in the paper's structure. Our study adds to the development of web security and safeguards consumers from potential danger by addressing the drawbacks of existing ways and utilizing a more sophisticated and exact strategy.

II. LITERATURE REVIEW(LR)

One of the biggest dangers facing internet users everywhere is malware. Identifying and avoiding dangerous web pages is so crucial. Blacklisting, static analysis, and dynamic analysis are the three methods that research practitioners advise using to spot harmful online pages. A number of techniques have been covered in order, and each strategy has specific aims to meet. For instance, Tao et al.[3] introduced a unique framework for automatically determining whether online pages are dangerous or safe using supervised machine learning techniques. Based on certain characteristics, the websites were classified as malicious or not. The dataset Proceedings of the International Conference on Intelligent Computing and Control Systems was used to gather benign web pages (ICICCS 2020).

Aldwairi[4] et al. presented a novel, lightweight self-learning method for categorising malicious web pages based on the properties defined in the A MALURL framework. In order to teach classifiers to recognize fraudulent web sites, they created a Genetic Algorithm (GA). The dataset utilized Alexa for good websites and Phis Tank for bad websites. It was discovered that the typical system precision was 87%.

In order to distinguish between known and unidentified malicious web pages, Yoo and colleagues [5] suggested two distinct detection techniques: anomaly detection and misuse detection. The false positive rate was high, at 30.5%, despite the detection allowance being relatively max up to 98.9%. They used the RafaBot dataset and the WEKA tool to carry out their experiment.

Hwang[6] et al. used a machine learning method known as "Adaptive SVM(aSVM)". Because of its flexibility to adapt, the aSVM can handle new training data. To lessen the likelihood of misclassifying fresh web pages is the goal of aSVM.

Yue et al.[7] proposed a malicious webpage classification method that utilizes 30 features and two machine learning algorithms, K-NN and SVM. KNN outperformed SVM in terms of results. To find malicious websites and specific threat types, two classification models were applied.

SpiderNet was created by Krishnaveni et al. [8] to aid in the identification of harmful websites. MatLab was used to develop the tool, and it made use of three feature sets: those for handling redirects, for handling JavaScript and common features, and for handling cross-site scripting [9] attacks. The software incorporates two machine learning classifiers—ELM and multi-SVM. The experimental results showed that the ELM classifier had a greater percentage of success (96.62%) than the multi-SVM classifier (93.22%).

In order to discover and examine novel harmful URLs, Sun et al. [10] introduced a technology called Automated Blacklist Generation (AutoBLG). In order to accomplish its goals, the framework made use of a number of methods, including URL Filtering, URL Expansion, and URL Verification. Using these methods, AutoBLG successfully completed its objectives.

In [11], a novel method for identifying fraudulent websites was proposed by Wang et al. Combining static and dynamic analysis into a single strategy. During static analysis, we were able to extract features from websites and use them to teach a classifier to determine whether or not a website included dangerous content. In contrast, web pages are run via the browser engine and their activity is monitored in a process known as dynamic analysis. Combining the benefits of the two methods, the hybrid approach outperformed the other two approaches despite the fact that the initial approach resulted in a high number of false negatives, while the latter required greater computational resources.

WebMon, created by Kim et al. [12] was 7.6 times faster than traditional appliances at detecting changes to web pages. To identify harmful websites, a model was proposed using WebKit-2, ML, and YARA. Additionally, a call tree algorithm for building a malicious redirection tree that could identify the malicious path was presented.

Altay et al. [13] introduced a supervised machine learning approach to classify web pages based on keyword density in a context-sensitive manner. The study employed various algorithms, such as SVM, maximum entropy, and ELM, to achieve the classification task.

In conclusion, there are several ways to go about identifying dangerous websites. While each technique has pros and cons, the main flaw in most methods is the reliance on large sample sizes to produce reliable results. These samples can contain tens to thousands of examples, making their collection and analysis labor- and resource-intensive. Therefore, it would be advantageous to create effective methods that use fewer samples for real-time detection of malicious websites.

III. METHODOLOGY

Our suggested method for detecting malicious web pages is based on a fresh web site classification scheme created to address the limitations of earlier research. Figure 1 shows how the system, which uses URL features to spot rogue websites, works.

Our method starts with gathering a collection of websites from a variety of sources, some of which were determined to be malicious and others considered safe. The data in this dataset is then condensed in order to be filtered and cleaned by choosing the most pertinent attributes from a total of 21 attributes. In step 3, we created a brand-new dataset with 450176 records and 16 URL features. After that, the dataset is manually split into two sets: a training set (180070 records) and a testing set (270105 records)[14]. To create a Machine Learning (ML) model, machine learning classifiers are trained using the training set in step 4. XGBoost, Logistic Regresiion, Decision Tree, Random Forest, Naive Bayes, K Nearest Neighbours, Support Vector Machine, ANN, Adaboost, Gradient Boosting are some of the classifiers employed. The flowchart for above methodology is shown in Figure 1.

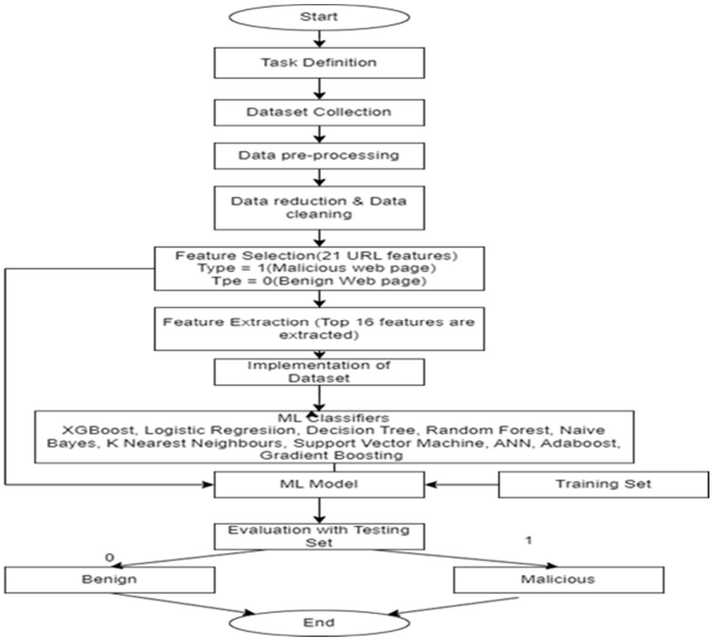

Start
Task Definition
Dataset Collection
Data pre-processing
Data reduction & Data cleaning
Feature Selection(21 URL features)
Type = 1(Malicious web page)
Tpe = 0(Benign Web page)
Feature Extraction (Top 16 features are extracted)
Implementation of Dataset
ML Classifiers
XGBoost, Logistic Regresiion, Decision Tree, Random Forest, Naive Bayes, K Nearest Neighbours, Support Vector Machine, ANN, Adaboost, Gradient Boosting
ML Model
Training Set
Evaluation with Testing Set
0
1
Benign
Malicious
End

Fig.1 A Proposed Strategy for Detecting Malicious Web Pages

a) Dataset Selection and Cleaning

The dataset chosen for training the machine learning algorithms has a significant impact on the classification quality. We gathered a dataset from the Kaggle database that contains 450176 records of both malicious and benign websites for our suggested method. The visualization for the benign and malicious is shown in figure 2. However, we reduced the dataset by choosing only pertinent attributes out of a total of 21 attributes in order to enhance the performance of our approach. The dataset was then manually split into two sets, one for training with 180070 records and the other for testing with 27015 records. We trained machine learning classifiers on the training set, and we tested their effectiveness on the testing set. This procedure made certain that our method could accurately detect malicious websites without over- or under-fitting.

The testing data along with the url and label is provided in Figure3.

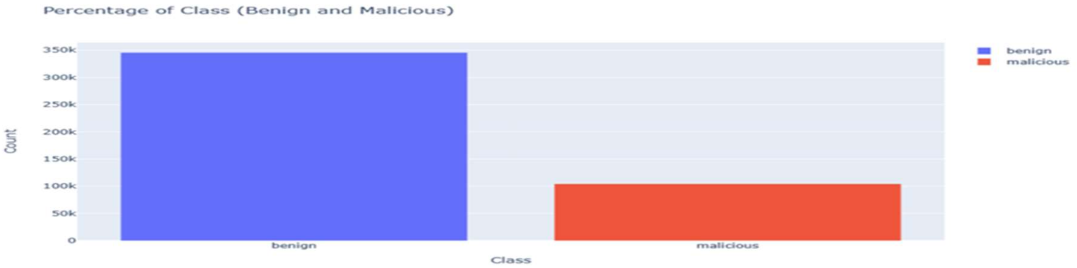
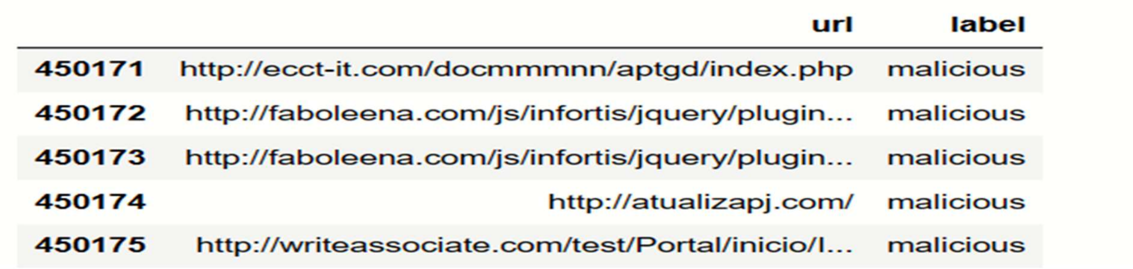


Fig.2 The percentage of Benign and Malicious websites

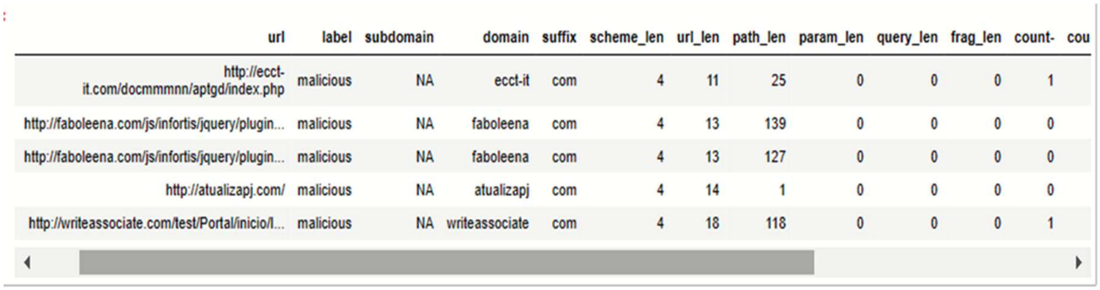


	url	label
450171	http://ecct-it.com/docmmmnn/aptgd/index.php	malicious
450172	http://faboleena.com/js/infortis/jquery/plugin...	malicious
450173	http://faboleena.com/js/infortis/jquery/plugin...	malicious
450174	http://atualizapj.com/	malicious
450175	http://writeassociate.com/test/Portal/inicio/l...	malicious

Fig.3 A quick glance at the information we have collected

b) Feature Extraction

A crucial step in machine learning is feature extraction, which reveals patterns and connections between the input variables (features) and the target variable (the type of website). In our suggested method, we manually extracted seven crucial hosted and syntactical features from the website URLs. These characteristics include the URL's length, the number of slashes, the number of underscores, and hyphens. SOURCE-APP-PACKETS and REMOTE-APP-PACKETS, two essential features that were not utilised in prior methods, were also added. These characteristics distinguish malicious from benign websites significantly and are more likely to signal a risk to web pages. We made sure that our machine learning classifiers could correctly identify malicious websites by incorporating these features into our methodology. The top features are listed in figure 4.



url	label	subdomain	domain	suffix	scheme_len	url_len	path_len	param_len	query_len	frag_len	count-	cou
http://ecct-it.com/docmmmnn/aptgd/index.php	malicious	NA	ecct-it	com	4	11	25	0	0	0	1	
http://faboleena.com/js/infortis/jquery/plugin...	malicious	NA	faboleena	com	4	13	139	0	0	0	0	
http://faboleena.com/js/infortis/jquery/plugin...	malicious	NA	faboleena	com	4	13	127	0	0	0	0	
http://atualizapj.com/	malicious	NA	atualizapj	com	4	14	1	0	0	0	0	
http://writeassociate.com/test/Portal/inicio/l...	malicious	NA	writeassociate	com	4	18	118	0	0	0	1	

Fig.4 feature subset

c) Models for machine learning-based classification

Many different machine learning classifiers, such as Adaboost, XGBoost, Random Forest, Support Vector Machine, Nave Bayes, Logistic Regression, Decision Tree, K Nearest Neighbors, ANN, and Gradient Boosting, are used to detect malicious websites. For binary classification problems such as the detection of harmful websites, logistic regression excels. The ensemble learning method known as Random Forest mixes many decision trees to boost accuracy while classifying data. The Bayes theorem is the basis for the basic yet effective machine learning classifier known as Gaussian Naive Bayes. A powerful training method, Support Vector Machine can learn classification and regression rules from data[15]. The popular Decision Tree technique uses a tree structure to represent options and their potential outcomes. Classifying a data point according to how its neighbours are classified, K Nearest Neighbors (KNN) is a non-parametric approach to classification. The structure of the human brain serves as inspiration for Artificial Neural Networks (ANN), a type of machine learning model used for categorization tasks. To construct a robust classifier, Gradient Boosting combines the knowledge of several weak learners into a single one. Finally, Adaboost and XGBoost are two other widely used ensemble algorithms that pool the strengths of numerous weak classifiers into a single robust one.

IV. PROPOSED APPROACH

The process we implemented to identify malicious web pages involved multiple critical steps, including dataset selection, feature extraction, and machine learning classifiers. Our approach aimed to provide accurate and efficient results to enhance web security for individuals and organizations.

To start, we focused on dataset selection. We used a dataset that contained 450176 records and 16 URL features, which was carefully selected to ensure that it was unbiased and free of inconsistencies. To create our training and testing datasets, we divided the original dataset manually, allocating 180,070 records for training and 270,105 records for testing to allow us to train our machine learning classifiers on a diverse range of examples while also testing the accuracy of our approach on unseen data as described in figure 1.

Finally, we utilized a combination of machine learning classifiers, including Adaboost, XGBoost, Random Forest, Support Vector Machine[16], Nave Bayes, Logistic Regression, Decision Tree, K Nearest Neighbors, ANN as shown in figure 5, and Gradient Boosting. Each classifier has unique strengths and weaknesses, and by combining them, we achieved higher accuracy than any single classifier alone. For example, Adaboost gave us the highest accuracy of 92%. The classifiers were trained on the training dataset and then evaluated on the testing dataset, indicating that our approach was effective in accurately identifying malicious web pages.

Overall, our approach to identifying malicious web pages involves a combination of careful dataset selection, feature extraction, and machine learning classifiers, making it an efficient and accurate way to identify malicious websites.

w

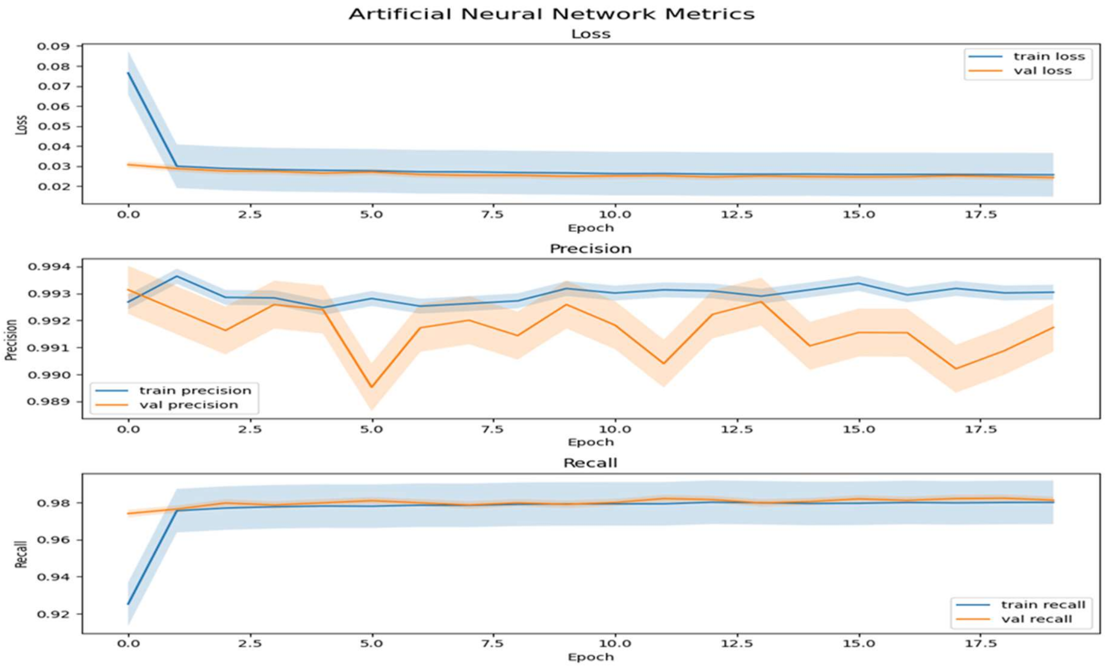


Fig.5 Metrics for gauging Artificial Neural Network effectiveness

V. RESULTS

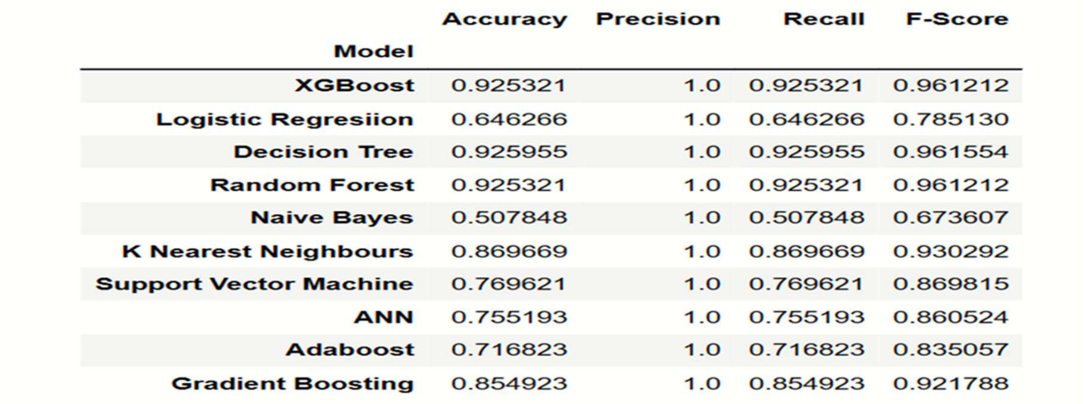
To assess the effectiveness of the suggested strategy for detecting malicious web pages, a number of machine learning classifiers, including Adaboost, XGBoost, Random Forest, Support Vector Machine, Nave Bayes, Logistic Regression, Decision Tree, K Nearest Neighbors, ANN, and Gradient Boosting, have been implemented and tested. Jupyter Notebook enables the presentation of code flow in a more visually appealing and comprehensible manner by providing support for popular data science libraries such as Matplotlib, Scikit-Learn, and Pandas, as well as for tables, plots, and markup languages.

With reported accuracy rates of up to 91%, previous studies on the detection of malicious web pages have produced promising results using machine learning algorithms like Random Forest (RF). However, we have added two more classifiers: AdaBoost, to further increase the accuracy of our method.

Using the same dataset as the earlier studies, we trained and tested our system, and we found that it was even more accurate than before. In particular, the RF classifier still had a 91% accuracy rate. The accuracy rate was 92.9% when AdaBoost was added.

These outcomes show the potency of our strategy and the potential advantages of using multiple classifiers to enhance the precision of malicious web page detection. We are able to develop a more effective and dependable system for identifying malicious web pages by utilizing the advantages of various classifiers.

In below Figure 6, a comparison of the accuracy of various classifiers is shown. According to the findings, Gaussian Naive Bayes only achieves a 50% accuracy rate, which is significantly less than that of the other classifiers. The accuracy of the logistic regression (LR) classifier is 64%, compared to 76% for the SVM classifier. The accuracy of the decision tree is 91%. KNN has an 86% accuracy rate. These findings show that the suggested strategy, which relies on machine learning classifiers and URL features, is an effective and efficient way to recognize malicious web pages. The comparison chart shown in figure 7.



Model	Accuracy	Precision	Recall	F-Score
XGBoost	0.925321	1.0	0.925321	0.961212
Logistic Regresiion	0.646266	1.0	0.646266	0.785130
Decision Tree	0.925955	1.0	0.925955	0.961554
Random Forest	0.925321	1.0	0.925321	0.961212
Naive Bayes	0.507848	1.0	0.507848	0.673607
K Nearest Neighbours	0.869669	1.0	0.869669	0.930292
Support Vector Machine	0.769621	1.0	0.769621	0.869815
ANN	0.755193	1.0	0.755193	0.860524
Adaboost	0.716823	1.0	0.716823	0.835057
Gradient Boosting	0.854923	1.0	0.854923	0.921788

Fig.6 A Comparison of the classification accuracy percentages among different classifiers

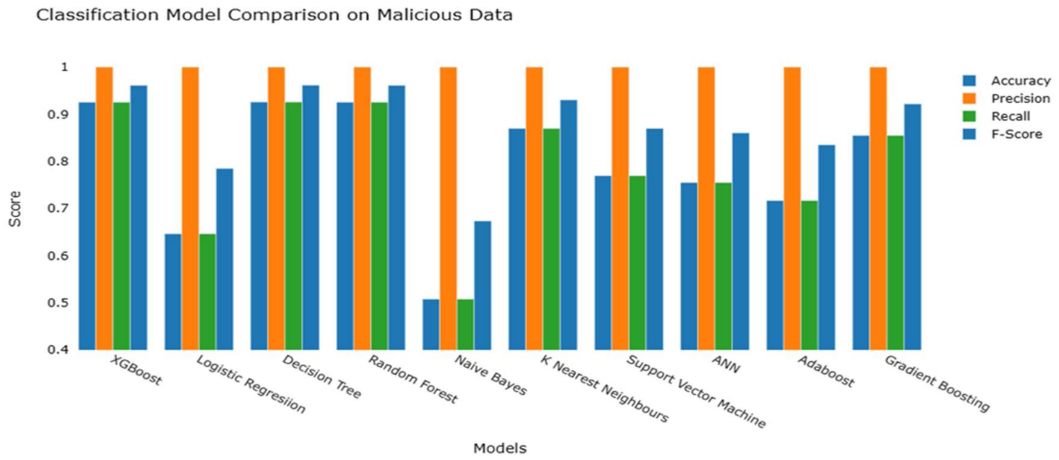


Fig.7 A comparison models on Malicious Data

VI. DISCUSSION

The main objective of this research study was to create a classification system that uses URL features to identify malicious internet sites, has been successfully addressed by the experimental findings of this study. Our study's findings show that machine learning algorithms, particularly Adaboost, can accurately classify web pages as malicious or benign with a 92% accuracy rate. After the discussion, the conclusion can be reached easily because the results have been carefully interpreted and examined. Each result's interpretation has been given, along with a performance comparison of the various classifiers and an expansion of feature sets for future research.

Our findings are consistent with other investigations and with earlier research studies that have used machine learning algorithms to identify malicious web pages. However, the choice of feature sets and the particular algorithm employed make a difference. The limitations of our study must also be acknowledged, particularly the fact that we only used a small number of URL-based features and did not use additional data sources. To improve the performance of the classifier even more, future research can go further on these restrictions. Overall, this study has significantly advanced cybersecurity and has produced encouraging findings for the identification of malicious web pages.

VII. CONCULUSION

In conclusion, our research on detecting malicious web pages using machine learning algorithms has produced encouraging outcomes. With a higher accuracy of 92%, We developed a new web site classification system based on URL features and used it to train multiple classifiers, including Adaboost, to identify malicious web pages

Our method not only outperformed earlier research, but it also showed how useful it is to use a limited number of pertinent URL features for classification. The presentation of our code flow was made more appealing and understandable by the use of Jupyter Notebook for coding and testing.

To further improve the performance of our classifiers, we intend to expand our feature set and examine various data sources in upcoming work. It is essential to keep looking into fresh, cutting-edge approaches to identify and stop malicious web activity as the cybersecurity landscape changes. Our study advances the work being done in this ongoing project.

In order to find areas for additional improvement, we intend to further examine the effectiveness of various classifiers and feature sets in future work. In order to improve the accuracy of our system, we also intend to investigate additional data sources outside of URL features, such as content-based features. We believe that by making these efforts, our strategy can support ongoing efforts to strengthen cybersecurity and safeguard internet users from malicious web pages

VIII. REFERENCES

[1] Eshete, Birhanu, Adolfo Villafiorita, and Komminist Weldemariam discuss effectiveness and efficiency issues in malicious website detection in the First SysSec Workshop 2011.

[2] Urcuqui, Christian, Andres Navarro, Jose Osorio, and Melisa García discuss machine learning classifiers to detect malicious websites in SSN 2017.

[3] Tao, Wang, Yu Shunzheng, and Xie Bailin propose a novel framework for learning to detect malicious web pages in the International Forum on Information Technology and Applications 2010.

[4] Aldwairi, Monther, and Rami Alsalman present Malurls, a lightweight malicious website classification based on URL features, in the Journal of Emerging Technologies in Web Intelligence 2012.

[5] Yoo, Suyeon, Sehun Kim, Anil Choudhary, O. P. Roy, and T. Tuithung introduce a two-phase malicious web page detection scheme using misuse and anomaly detection in the International Journal of Reliable Information and Assurance 2014.

[6] Hwang, Young Sup, Jin Baek Kwon, Jae Chan Moon, and Seong Je Cho propose classifying malicious web pages by using an adaptive support vector machine in the Journal of Information Processing Systems 2013.

[7] Yue, Tao, Jianhua Sun, and Hao Chen present fine-grained mining and classification of malicious web pages in the Fourth International Conference on Digital Manufacturing & Automation 2013.

[8] Krishnaveni, S., and K. Sathiyakumari propose SpiderNet, an interaction tool for predicting malicious web pages in the International Conference on Information Communication and Embedded Systems 2014.

[9] Ibrahim, M. Y. presents a machine learning approach to real-time XSS detection in 2017.

[10] Sun, Bo, Mitsuaki Akiyama, Takeshi Yagi, Mitsuhiro Hatada, and Tatsuya Mori introduce an approach to automating URL blacklist generation with a similarity search approach in IEICE Transactions on Information and Systems 2016.

[11] Wang, Rong, Yan Zhu, Jiefan Tan, and Binbin Zhou propose the detection of malicious web pages based on hybrid analysis in the Journal of Information Security and Applications 2017.

[12] Kim, Sungjin, Jinkook Kim, Seokwoo Nam, and Dohoon Kim introduce WebMon, a machine learning and YARA-based malicious webpage detection system, in Computer Networks 2018.

[13] Altay, Betul, Tansel Dokeroglu, and Ahmet Cosar propose context-sensitive and keyword density-based supervised machine learning techniques for malicious webpage detection in Soft Computing 2019.

[14] Sanjukta Mohanty, Arup Abhinna Acharya, Laki Sahu, Sunil Kumar Mohapatra(2020).Hazard Identification and detection using machine learning . In 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS).

[15] Gaikwad S, Nale P, Bachate R (2016) Survey on big data analytics for digital world. In: IEEE international conference on advances in electronics, communication and computer technology. Pune, pp 180–186

[16] Gabriel AD, Gavrilut DT, Alexandru BI, Stefan PA (2016) Detecting malicious URLs: a semi_x0002_supervised machine learning system approach. In: 18th ınternational symposium on symbolic and numeric algorithms for scientific computing. Timisoara, pp 233–239