International journal of engineering science and
management (IJESM).
ISSN: XXXX-XXXX.
https://doi.org/XX.XXXXX/XXXXXXXX/IJESM-XXXXXXXXX

Volume: 24, Issue 1, 91-103, October 2024.

HAZARD IDENTIFICATION AND DETECTION USING MACHINE LEARNING

Dr. P. Rama Koteswara Rao[1], Dr. Kiran kumar[2]
Professor, Department of ECE, NRI INSTITUTE OF TECHNOLOGY, Agiripalli, Andhra Pradesh, India, Email Id: dr.ramakoteswarao@gmail.com
Professor, Dept. of CSE, NIT Tadepalligudem, Email Id: ritikabateja@gmail.com

Received: 13 September 2024 Revised: 04 September 2024 Accepted: 3 October 2024 Published: 4 October 2024

ABSTRACT:

Web surfing is become an integral part of our daily life in the modern world. But along with this flexibility comes the possibility of visiting rogue websites that might infect our devices with malware and steal our private information. Regrettably, our reliance on conventional cybersecurity measures such as firewalls and antivirus software fall short in adequately safeguarding us against these emerging threats. Consequently, there is an urgent demand for a more advanced and practical model capable of consistently discerning safe from perilous websites. In response to this imperative, we have developed a novel classification system that scrutinizes and detects features associated with URLs, employing a diverse array of machine learning classification algorithms. These algorithms encompass Adaboost, XGBoost, Random Forest, Support Vector Machine, Naive Bayes, Logistic Regression, Decision Tree, K Nearest Neighbours, Artificial Neural Networks (ANN), and Gradient Boosting. Notably, Adaboost, XGBoost, and Support Vector Machine have garnered recognition within the research community for their effectiveness in identifying fraudulent websites. Our primary objective is to construct a system capable of reliably ascertaining the intent of a website, specifically whether it harbours malicious intentions towards its users. To achieve this, we intend to leverage bagging and boosting techniques to train these classifiers subsequent to extracting the requisite attributes from the websites. We will then subject our method to rigorous testing on a substantial dataset comprising web pages, facilitating a comprehensive comparison of its performance against competing methodologies. The results we present will illustrate the efficacy of our classification approach in pinpointing malevolent web pages. By furnishing a swift and precise means of identifying and analysing hazardous websites, our proposed strategy has the potential to significantly enhance web security and safeguard users. The implications of this study's findings could extend far beyond, ushering in profound changes in the realm of cybersecurity.

KEYWORDS – Machine Learning, Naïve Bayes, Logistic Regression, Decision Tree, KNN, SVM.

1. INTRODUCTION:

The widespread adoption of internet banking, online shopping, bill payment, e-learning, and various other services has become increasingly prevalent among consumers, thanks to the rapid expansion of the web. Individuals now rely on browsers or web applications to access these services. However, the growing capabilities and functionalities of browsers present a significant security and privacy risk for personal and sensitive information [1].

Inexperienced users are vulnerable to exploitation by malicious actors with a simple click on a malicious website, allowing these actors to leverage webpage vulnerabilities and gain unauthorized remote access to the user's system. Given the continuous expansion of the internet, it is imperative to ensure accurate identification of websites. To address these concerns, blacklisting services have been integrated into web browsers, despite their inherent limitations [2].

This paper delves into a self-learning approach for online page classification. We employ four machine learning classifiers to categorize internet sites into two distinct categories: benign and hazardous web pages [3].

To underscore the significance of our proposed approach, we aim to elucidate the deficiencies inherent in current blacklisting systems [4].

We contend that these systems often harbor inaccuracies and lag behind the evolving tactics employed by cybercriminals. Our overarching goal in this study is to formulate a more dependable strategy geared towards safeguarding users through precise identification and preemptive avoidance of harmful online activities [5].

Our inquiry centers around the feasibility of employing machine learning classifiers to differentiate between secure and malicious websites [6].

Additionally, we seek to assess the effectiveness of the proposed approach in detecting malicious content on the web [7].

In light of these considerations, it becomes imperative to accurately discern web pages within the ever-expanding online landscape. Although blacklisting services were introduced in web browsers to address these issues, they suffer from various drawbacks, notably inaccurate listings [8].

This paper delves into a self-learning approach for website classification, utilizing a constrained set of features. We classify websites into two categories—malicious and benign—using four machine learning classifiers [9].

Given that uninformed users have limited knowledge about various types of malwares, they can easily fall victim to intruders with a mere click on malicious websites [10].

This enables attackers to exploit vulnerabilities within the website and inject payloads to gain remote access to the victim's system. In consideration of these factors, the precise identification of websites within the ever-expanding online space is of paramount importance. To address these challenges, blacklisting services have been integrated into web browsers [11].

Nevertheless, they come with their own set of limitations, including the issue of inaccurate listings and the basic system architecture is displayed in below fig 1.

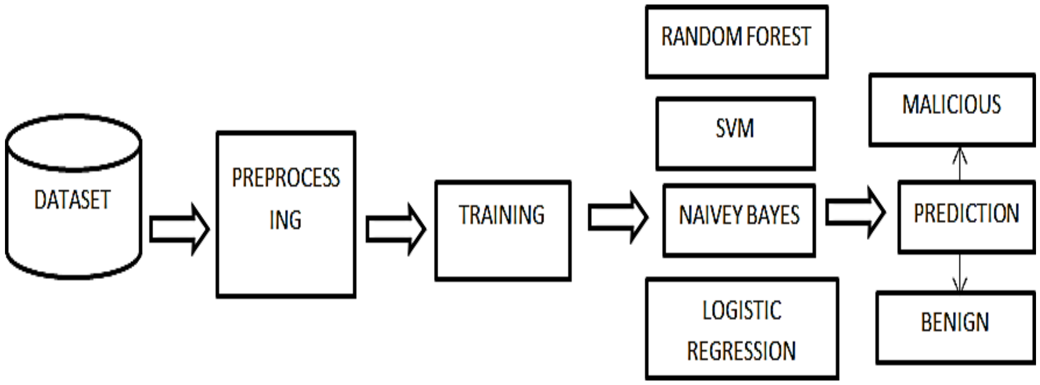


Figure.1. Example Figure

Distinguishing proof of hazards is a stage during the time spent deciding whether a specific situation, object, element, and so forth, can possibly be unsafe. The whole methodology is as often as possible alluded to as gamble with appraisal: Decide the gamble variables and risks that could be unsafe (peril distinguishing proof). Designing controls, replacement, regulatory controls, individual defensive gear, and disposal are recorded arranged by expanding adequacy [12].

To best safeguard laborers, you'll much of the time need to blend control draws near.
The seven average dangers at work are:

- Dangers to somewhere safe and secure.
- Natural dangers.
- Risks that are ergonomic
- Risks that are physical.
- Synthetic dangers.
- Hierarchical dangers at work.
- Threats to the climate.

2. LITERATURE SURVEY:

The danger presented by pernicious sites to the security of the web is notable. Drive-by download assaults are started by pernicious sites fully intent on assuming control over a client's PC and involving it for illegal purposes. rebel malware executables can be downloaded and introduced naturally by essentially visiting a maverick site. In this exploration, we offer a special technique in view of regulated AI to consequently recognize site pages as harmless or unsafe. Our strategy just purposes HTTP meeting data — like solicitation and reaction spaces and HTTP meeting headers — to distinguish fake sites. We can distinguish 92.2% of the pernicious pages with a low misleading positive pace of 0.1% utilizing a corpus of 50,000 harmless and 500 malignant sites [1].

At the point when a casualty visits a maverick site, her PC becomes contaminated, permitting programmers to take significant information, reroute her to other malevolent sites, or undermine her framework to send off additional assaults. The location of vindictive sites can be helped by the ongoing methodologies, however there are still issues that should be settled to really and effectively channel pages from the wild, cover a large number of malignant qualities to catch the 10,000-foot view, ceaselessly develop website page highlights, consolidate highlights in a deliberate way, and consider the ramifications of component values for site page portrayal. Also, the examination and identification methods should be adaptable and versatile to represent unavoidable changes in the danger scene. With an emphasis on more extensive element space and assault payloads, adaptability in strategies to oblige changes in noxious qualities and site pages, and in particular, certifiable convenience of methods in safeguarding clients against vindictive sites, we feature our continuous endeavors in this position paper to dissect and identify pernicious sites in a compelling and proficient way [2].

With the multiplication of different types of assaults Online, World Wide Web (WWW) is transforming into a dangerous day to day action. Various cheats, phishing plans, fraud, SPAM business, and infections begin for the most part from sites. However, popup blockers, boycotts, and program expansions alone can't completely shield clients. That calls for fast, exact frameworks that can recognize recently hurtful substance. We give MALURLs, a lightweight framework to recognize noxious sites on the web in view of host and URL lexical properties. To decide whether an objective site is vindictive or harmless, the framework utilizes a probabilistic model called the Credulous Bayes classifier. To increment order exactness and speed, it adds new highlights and uses a hereditary calculation for self-learning. Utilizing GA changes, a little dataset is accumulated and extended to empower speedy and low-memory framework learning. A few solid web-based sources are utilized to naturally gather and approve a testing dataset that is completely free. The precision of the calculation is 87% by and large [3].

Ongoing advancements in PCs and PC networks have improved the probability that programmers might involve site pages for destructive purposes. This paper planned and built a crossover pernicious URL discovery framework joining Naive Bayes and Decision Tree. The structure, an identification model, is created on three principal parts: include extraction, grouping modules, and a characteristic interaction. Its goal is to classify site pages as either hazardous or harmless. 12 HTML archive highlights were removed from every URL utilizing the JSoup Web Content component extractor. PhishTank and Alexa rating sites were assembled for the URL corpus, comprising of 3000 harmless and 355 malignant pages. The framework effectively arranged URLs as harmless and pernicious with 96.6% and 83.7% precision, separately, as per characterization tests directed in the WEKA climate. Interestingly, the Cross-breed URL Location Model, Choice Tree, and Guileless Bayes had Identification Rates (DR) of 93.1%, 83.1%, and 66.1%, individually, and False Positive Rates (FPR) of 6.7%, 16.9%, and 33.9%. Eventually, on the preparation and testing datasets, the Troupe Classifiers showed an exactness of 97.7% and 93.1%, separately [4].

We made 14 essential and 16 extra highlights to analyze a page as unsafe or harmless. The major components that we included were decided to represent the critical parts of a site page. The framework really recognizes harmless and malevolent pages by heuristically joining two crucial elements into one broadened include. These highlights can be utilized to prepare the support vector machine to order pages effectively. Given the quick advancement of pernicious sites, classifiers prepared on verifiable information may misclassify a few recently made pages. We chose to utilize a versatile support vector machine (an SVM) as a classifier to tackle this issue. In light of the support vectors, it obtained during its earlier learning meeting, the a SVM can quickly get new preparation information notwithstanding its current preparation set. Results from the examinations affirmed that the a SVM can arrange unsafe pages in a versatile way [5].

3. METHODOLOGY:

Our proposed method for identifying malicious websites is founded on a novel website classification methodology, meticulously designed to rectify the deficiencies observed in previous studies. Refer to Figure 1 for an illustration of how our system detects malicious websites by analyzing their URL characteristics. The initial step in our process involves assembling a collection of URLs from diverse sources, encompassing both malicious and safe ones [13].

Subsequently, we curate and streamline the dataset by selecting the most pertinent attributes from a total of 21. In the third step, we generate a new dataset comprising 450,176 records and 16 URL characteristics. This dataset is then manually partitioned into a training set, comprising 180,070 records, and a testing set, consisting of 270,105 records [14].

The fourth step entails the training of machine learning classifiers on the training set, facilitating the construction of a robust ML model [15].

Among the classifiers utilized are XGBoost, Logistic Regression, Decision Tree, Random Forest, Naive Bayes, K Nearest Neighbors, Support Vector Machine, Artificial Neural Network, Adaboost, and Gradient Boosting. Figure 2 provides a visual representation of the outlined procedure.

Drawbacks:

- They require tens, hundreds, or even a sizable number of models to find their responses.
- These strategies find it challenging to get preliminaries of numerous things, and they miss the mark on strategy for recognizing unlawful URL diverts, which is a powerful interaction.

The work that is suggested in the study is a technique for recognizing and seeing dangers, which infers a ML strategy. As per the innovators, you can recognize among protected and hazardous site pages exclusively by checking the URL out. They propose three strategies for recognizing possibly destructive sites: boycotting, static investigation, and dynamic assessment.

Benefits:

- The proposed methodology utilizes ML procedures, which can secure from information and get better after some time. This makes the framework more flexible to wagers with that change throughout a lengthy time.
- The proposed strategy involves three methodologies for finding disastrous site pages: boycotting, static assessment, and dynamic appraisal. These strategies can work on the exactness of spotting and tracking down chances.

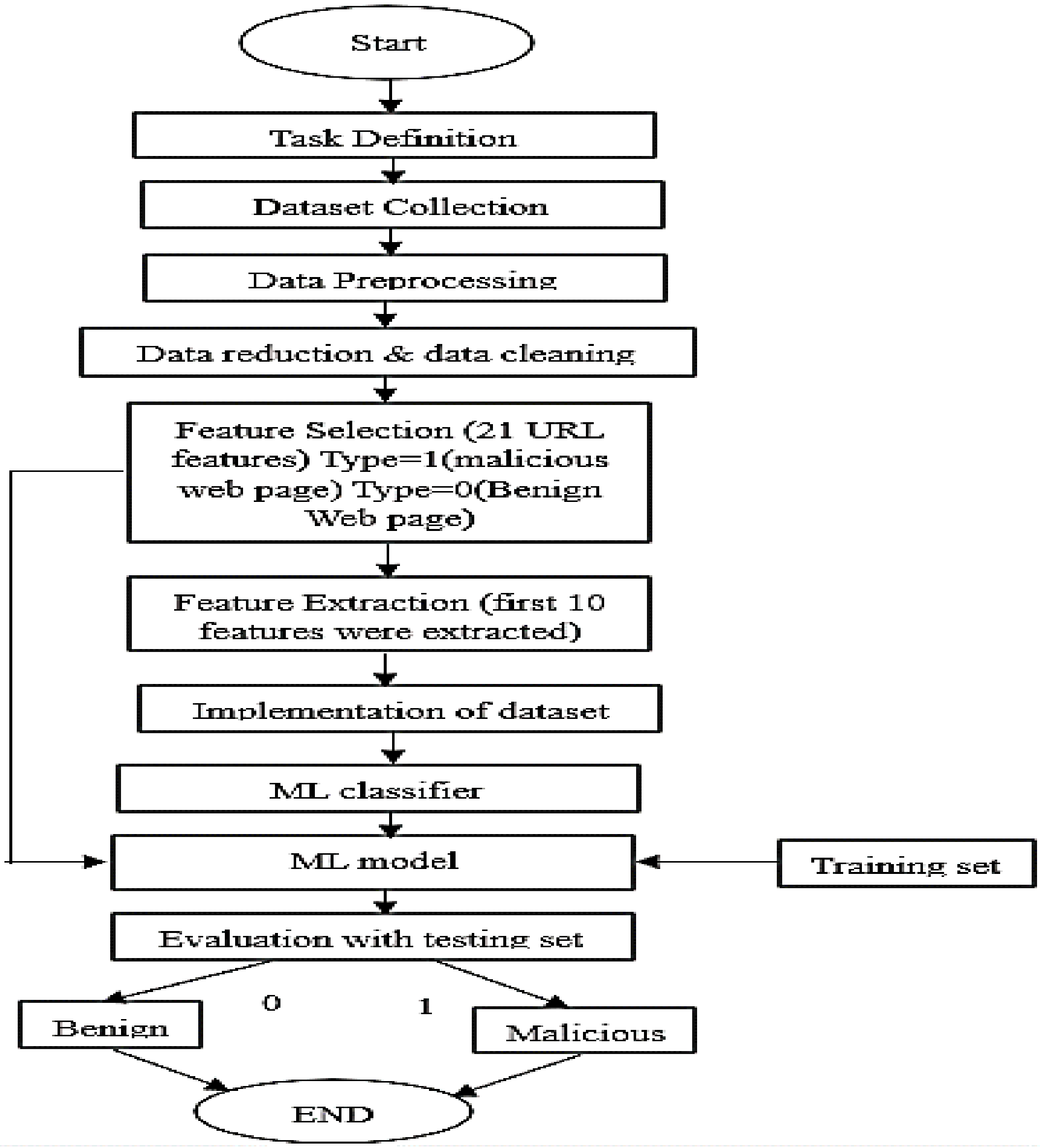

Start
Task Definition
Dataset Collection
Data Preprocessing
Data reduction & data cleaning
Feature Selection (21 URL features) Type=1(malicious web page) Type=0(Benign Web page)
Feature Extraction (first 10 features were extracted)
Implementation of dataset
ML classifier
ML model
Training set
Evaluation with testing set
0
1
Benign
Malicious
END

Figure 2 An Approach to Identifying Threatful Websites

Dataset Selection and Cleaning: The accuracy of a machine learning algorithm's classifications relies heavily on the quality of the database used for training the system. To evaluate our proposed methodology, we conducted tests on a dataset created from the Kaggle database, comprising a total of 450,175 entries representing both dangerous and safe websites [16].

Figure 3 provides a graphical representation of scenarios depicting safe and harmful websites. To enhance the efficiency of our approach, we refined the dataset by selecting only three out of the 21 available attributes. Subsequently, we manually divided the dataset into two distinct sets: a training set comprising 180,070 entries and a testing set encompassing 27,015 records. Machine learning classifiers were then trained using the training set and evaluated on the testing set. This method assured that our approach could properly identify dangerous websites without requiring a disproportionate amount of over- or under-fitting. Figure 4 includes the testing data, together with the label and URL, for your convenience.

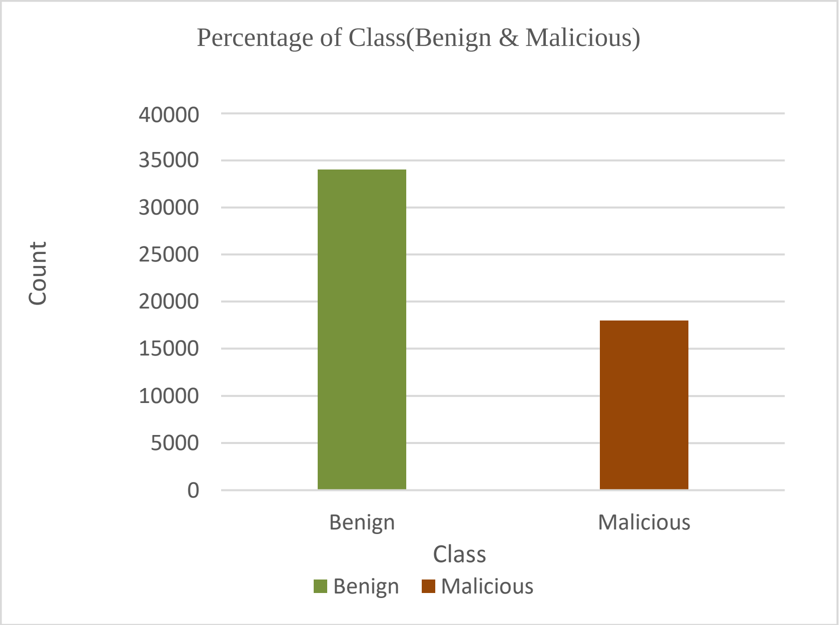
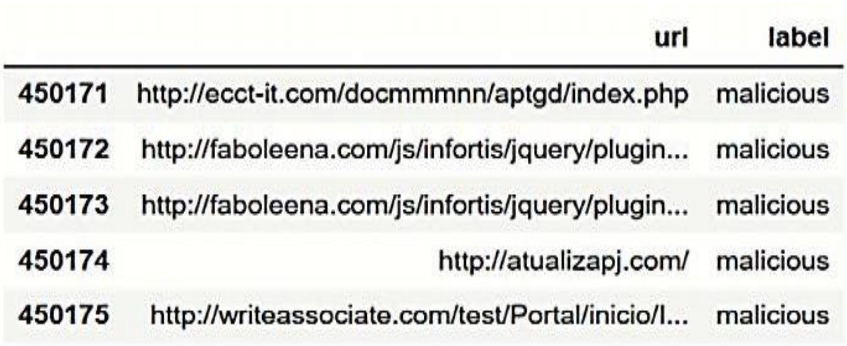


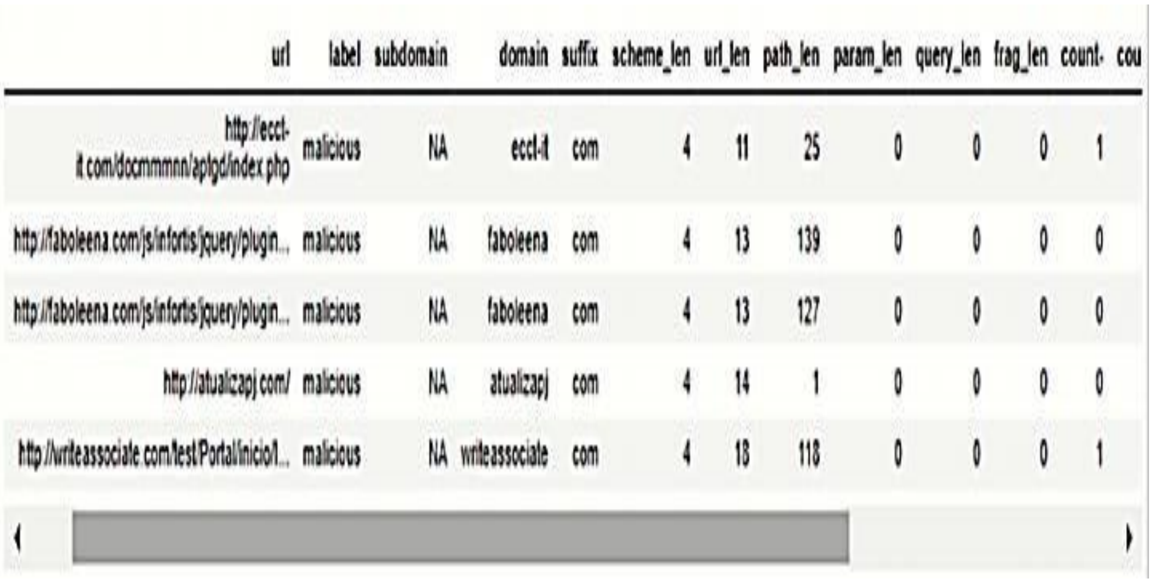
Figure 3: The websites of Benign & Malicious.



	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

Figure 4: A summary of the data we've gathered.

Feature Extraction: Machine learning relies heavily on a process called feature extraction, which establishes relationships and correlations between the information used in training and the final outcome, in this case the category of website. In our proposed procedure, seven vital hosted and syntactical aspects of the website URLs were retrieved manually. These elements include the length of the URL, the presence or absence of slashes, underscores, and hyphens, and the total amount of these characters. Essential functionality such as SOURCE-APP-PACKETS and REMOTE-APP-PACKETS that were previously underutilized have also been included. These traits are more indicative of a potential threat to web pages and help distinguish malicious from benign websites. By adding these characteristics into our approach, we ensured that our machine learning classifiers could accurately recognize fraudulent websites. Figure 5: Summary of Principal Features and the comparison among the classifiers are displayed in table 1 in below & then its graphical representation is displayed in Fig 6.



	url	label	subdomain	domain	suffix	scheme_len	url_len	path_len	param_len	query_len	frag_len	count-	cou
	http://eccl-it.com/docmmmn/aplgd/index.php	malicious	NA	eccl-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/1...	malicious	NA	writeassociate	com	4	18	118	0	0	0	1	

Figure 5: Feature subset.

Implementation: Logistic Regression: This factual technique predicts a twofold result (yes or no) by using verifiable perceptions of an information assortment. By examining the connection between at least one current free factor and the reliant variable, a strategic relapse model predicts the last option. For example, a calculated relapse might be utilized to anticipate on the off chance that a secondary school candidate would be acknowledged into a specific college or on the other hand assuming a political up-and-comer would win or lose a political decision. Basic choices between two choices are made conceivable by these paired results.

KNN: Likewise alluded to as k-NN or KNN, the k-nearest neighbors' technique is a non-parametric directed learning classifier that utilizes nearness to give expectations or characterizations on the gathering of a solitary data of interest.

SVM: A managed ML model called a support vector machine (SVM) utilizes characterization methods to resolve two-bunch grouping issues. In the wake of giving a SVM model arrangements of marked preparing information for each class, they can order new text.

Decision tree: Utilizing a stretching instrument, a choice tree is a diagram that shows generally potential results for a given information. Decision trees can be made the hard way, with specific programming, or with a graphical application. Decision trees can assist with centering conversations when a gathering needs to settle on a decision.

Gaussian Naive Bayes, or GNB, is a machine learning (ML) grouping calculation that depends on a probabilistic methodology and Gaussian dissemination. As indicated by Gaussian Naive Bayes, each boundary — likewise alluded to as a component or indicator — can freely foresee the result variable. Random Forest Classifier: comprises of countless individual choice trees that participate to frame an outfit. Each tree in the random forest produces a class expectation; our model purposes the class that gets the best votes to decide its gauge.

4. RESULTS:

Table 1: The comparision of classification Accuracy between various classifiers.

Model	Accuracy	Precision	Recall	F-score
XG-Boost	0.925432	1.0	0.914785	0.912674
Logistic regression	0.646342	1.0	0.689563	0.734159
Decision tree	0.928765	1.0	0.990345	0.934156
Random Forest	0.976438	1.0	0.978421	0.978129
Naive Bayes	0.593752	1.0	0.590531	0.667123
KNN	0.890316	1.0	0.812567	0.990124
SVM	0.789492	1.0	0.789347	0.878318
ANN	0.789591	1.0	0.778235	0.890123
Ada Boost	0.709673	1.0	0.756783	0.874321
GB	0.894821	1.0	0.845189	0.978125

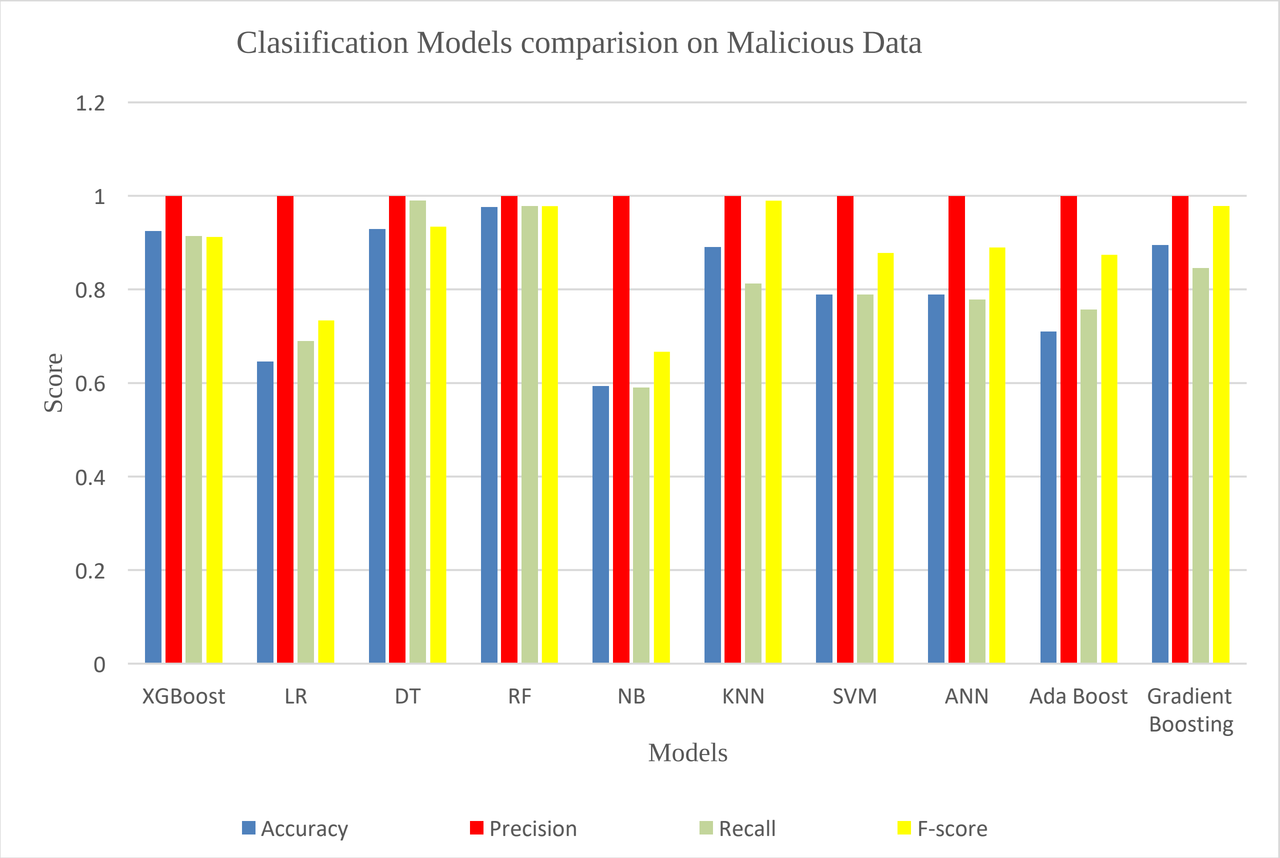


Figure 6: A model on Malicious data.

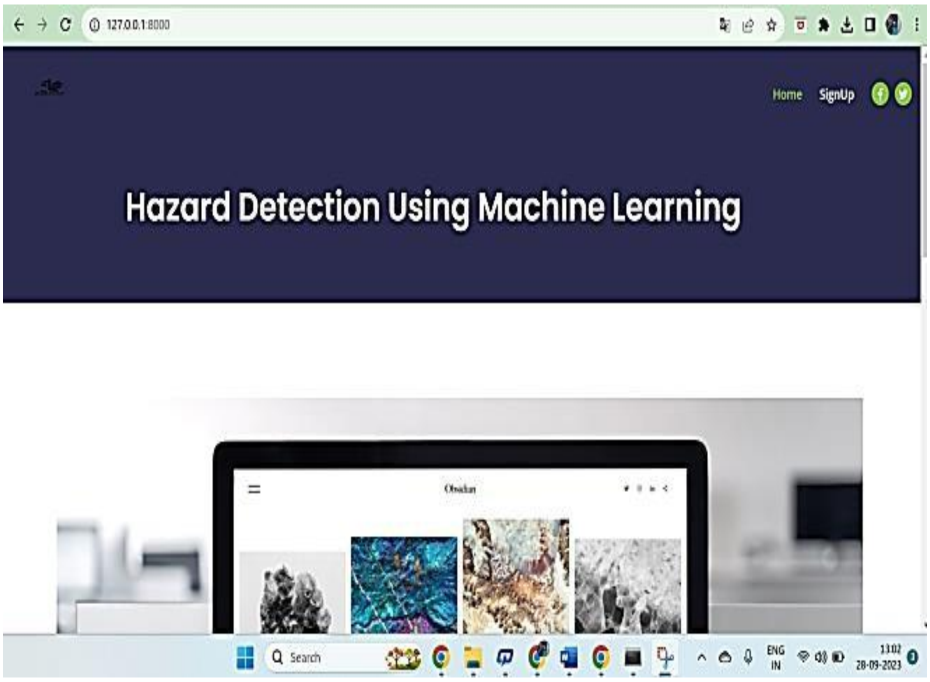


Figure 7 Home Page

The above screen shows the Home page for our proposed model and then the Fig 8 displays the registration page and then the we have to login to the website and it is displayed in the Fig 9.and then we can upload the url and it is displayed in Fig 10.the fig 11 and Fig 12 displays the prediction results of our proposed approach.

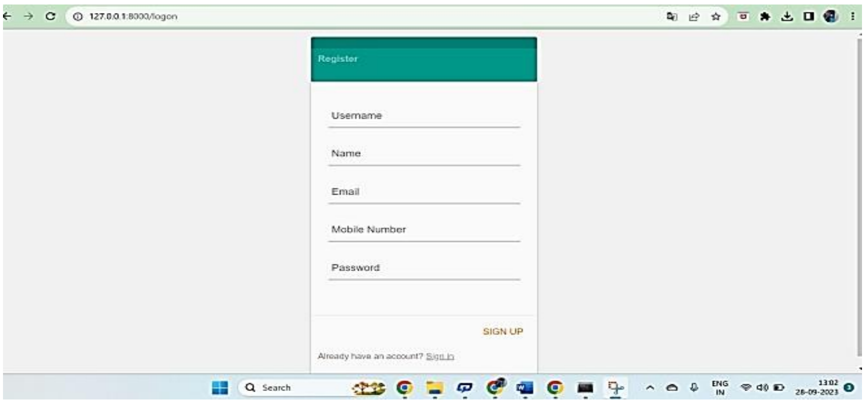
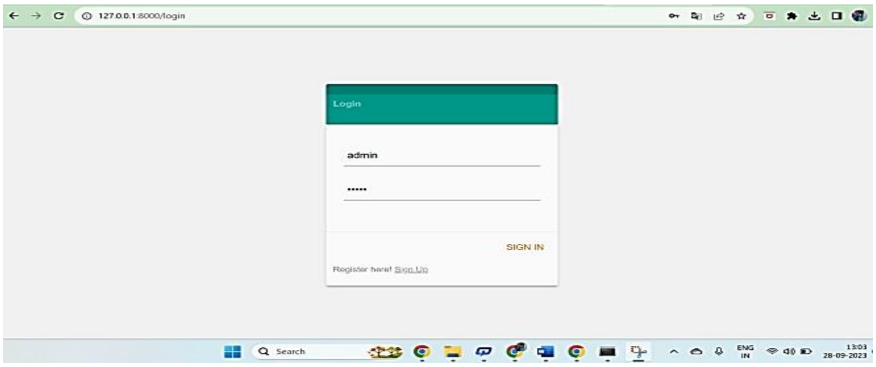


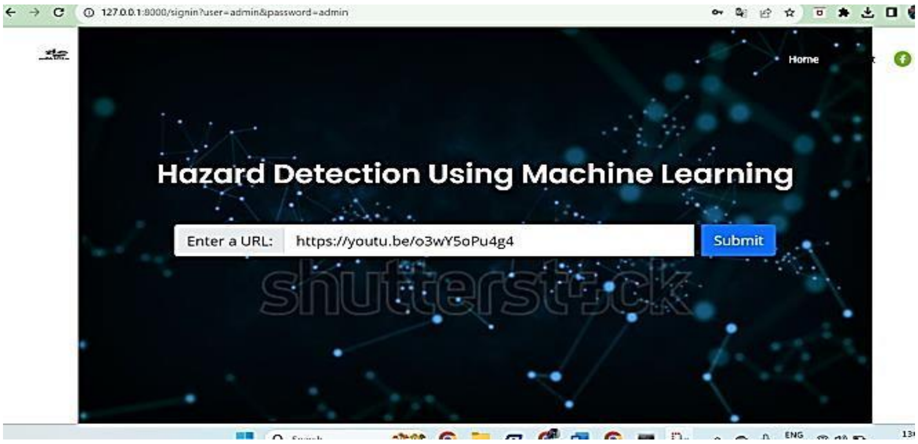
Figure 8 Registration Page



admin

•••••

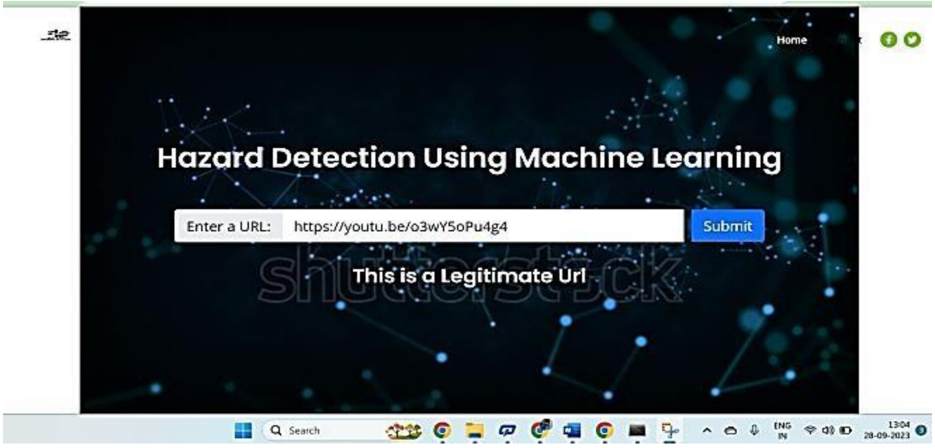
Figure 9 Login Page


127.0.0.1:8000/signin?user=admin&password=admin

Hazard Detection Using Machine Learning

Enter a URL: https://youtu.be/o3wY5oPu4g4

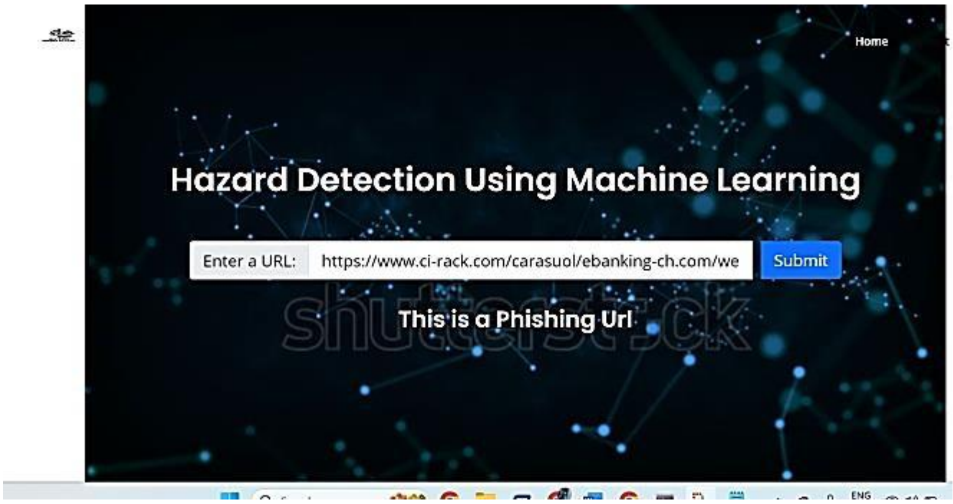
Figure 10 Upload URL


Hazard Detection Using Machine Learning

Enter a URL: https://youtu.be/o3wY5oPu4g4

This is a Legitimate Url

Figure 11 Prediction Result


Hazard Detection Using Machine Learning

Enter a URL: https://www.ci-rack.com/carasuol/ebanking-ch.com/we

This is a Phishing Url

Figure 12 Prediction Result.

5. CONCLUSION:

The distinguishing proof of vindictive pages is a creating field in network protection. Regardless of whether various investigations on the subject of vindictive page discovery have been directed, these are very costly on the grounds that they require some investment and cash. In this review, we utilized ML methods to anticipate whether the web-based pages are hazardous or harmless by using a clever framework for grouping sites in view of URL credits. The decision tree utilized by the ML classifiers arrives at a more noteworthy exactness of 96%. The results of the preliminary show the adequacy of our methodology in distinguishing risky sites. To work on the presentation of the classifier, it has been wanted to extend the capabilities and dissect information from a few sources in future work. To work on the exhibition of the classifier, it has been wanted to extend the capabilities and break down information from a few sources in future work.

6. REFERENCES:

[1] Tao, Wang, Yu Shunzheng, and Xie Bailin. "A novel framework for learning to detect malicious web pages." In 2010 International Forum on Information Technology and Applications, vol. 2, pp. 353-357. Ieee, 2010.

[2] Eshete, Birhanu, Adolfo Villafiorita, and Komminist Weldemariam. "Malicious website detection: Effectiveness and efficiency issues." In 2011 First SysSec Workshop, pp. 123-126. IEEE, 2011.

[3] Aldwairi, Monther, and Rami Alsalman. "Malurls: A lightweight malicious website classification based on url features." Journal of Emerging Technologies in Web Intelligence 4, no. 2 (2012): 128-133.

[4] Yoo, Suyeon, Sehun Kim, Anil Choudhary, O. P. Roy, and T. Tuithung. "Two-phase malicious web page detection scheme using misuse and anomaly detection." International Journal of Reliable Information and Assurance 2, no. 1 (2014): 1-9.

[5] Hwang, Young Sup, Jin Baek Kwon, Jae Chan Moon, and Seong Je Cho. "Classifying malicious web pages by using an adaptive support vector machine." Journal of Information Processing Systems 9, no. 3 (2013): 395-404.

[6] Yue, Tao, Jianhua Sun, and Hao Chen. "Fine-grained mining and classification of malicious Web pages." In 2013 Fourth International Conference on Digital Manufacturing & Automation, pp. 616-619. IEEE, 2013.

[7] Krishnaveni, S., and K. Sathiyakumari. "SpiderNet: An interaction tool for predicting malicious web pages." In International Conference on Information Communication and Embedded Systems (ICICES2014), pp. 1-6. IEEE, 2014.

[8] Sun, Bo, Mitsuaki Akiyama, Takeshi Yagi, Mitsuhiro Hatada, and Tatsuya Mori. "Automating URL blacklist generation with similarity search approach." IEICE TRANSACTIONS on Information and Systems 99, no. 4 (2016): 873-882.

[9] Urcuqui, Christian, Andres Navarro, Jose Osorio, and Melisa García. "Machine Learning Classifiers to Detect Malicious Websites." In SSN, pp. 14-17. 2017.).

[10] Wang, Rong, Yan Zhu, Jiefan Tan, and Binbin Zhou. "Detection of malicious web pages based on hybrid analysis." Journal of Information Security and Applications 35 (2017): 68-74.74.

[11] Kim, Sungjin, Jinkook Kim, Seokwoo Nam, and Dohoon Kim. "WebMon: ML-and YARA-based malicious webpage detection." Computer Networks 137 (2018): 119-131.

[12] Altay, Betul, Tansel Dokeroglu, and Ahmet Cosar. "Context-sensitive and keyword density-based supervised machine learning techniques for malicious webpage detection." Soft Computing 23, no. 12 (2019): 4177-4191.

[13] website: http://jupyter.org/

[14] https://archieve.ics.uci.edu/ml/dataset/

[15] Ibrahim, M. Y. (2017). Real Time Xss Detection: A Machine Learning Approach.

[16] https://medium.com/thalus-ai/performance-metrics-forclassification-problems-in-machine-learning-part-ib085d432082b