Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 5, No.11, May : 2024

EFFECTIVE DETECTION OF MALICIOUS URL'S USING ENSEMBLE LEARNING TECHNIQUES

Mrs.S.Anusha Reddy, Assistant Professor, Dept. of CSE(Data Science),Sreyas Institute of Engineering ,Nagole ,Hyderabad
Kanchanapally Pranitha, Kandagatla Sahithi, Annreddy Akshara, Donuri Praneeth Reddy, UG Student, Dept.of CSE(Data Science),Sreyas Institute of Engineering ,Nagole ,Hyderabad

Abstract—One type of internet security issue that targets human weaknesses instead of software flaws is malicious websites. It can be defined as the practice of luring internet users in order to get private data, including passwords and usernames. Because malicious URLs are a serious threat to cybersecurity, it is important to use efficient prediction techniques to find them among a large number of URLs. This research presents a novel method to improve the prediction accuracy of harmful URLs by utilizing machine learning techniques. A serious and constantly changing danger to cybersecurity is malicious URLs. The vast number of URLs makes it difficult to reliably identify fraudulent ones. Current approaches might not be able to adjust to the ever-changing strategies that fraudsters use. Inaccurate URL classification can have negative effects on security and compromise data. A large collection of labeled URLs, containing both benign and malicious samples, is used for experimental evaluations. The ensemble model's performance is contrasted with that of standalone classifiers and conventional machine learning methods. The results show that in terms of accuracy, precision, recall, and F1-score, the ensemble technique performs better than standalone classifiers. Furthermore, the ensemble shows good generalization to previously unexplored data and resilience against adversarial attacks. In order to develop a prediction model, the study investigates several pairings of training strategies and classification approaches. But by investigating additional ensemble techniques like Voting Classifier, Random Forest, XG Boost, and Light GBM, we may improve the performance.
Keywords— *Malicious URLS, Cyber Security, benign , Machine Learning Techniques, ensemble techniques.*

I. INTRODUCTION

Cyber threats have increased as a result of the internet's exponential expansion in popularity, especially in the form of malicious URLs (Uniform Resource Locators). Links that direct viewers to websites that contain hazardous material, such as phishing pages, malware distribution networks, and fraudulent websites, are known as malicious URLs. Therefore, it is now crucial to detect and mitigate these dangerous URLs effectively in order to ensure cyber security. Static blacklisting and signature-based techniques are frequently used in traditional URL detection strategies, however they are unable to keep up with the quickly changing online threat landscape. As a result, the demand for more reliable and flexible methods that can precisely identify malicious URLs in real-time is rising[1]. URL identification problems may be solved with the use of ensemble learning, a potent machine learning method that combines several base classifiers to increase prediction performance. Comparing ensemble approaches to individual classifiers, they provide higher accuracy and resilience by utilizing the diversity of numerous classifiers and their combined decision-making abilities. Malicious URLs can be introduced by ensemble learning techniques, which combine several weak learners to produce a strong classifier. One strategy is to combine the predictions of several models trained on various subsetsor representations of the data using techniques like bagging, boosting, or stacking. To detect fraudulent URLs, these models can be trained using features that are extracted from URLs, such as lexical features, content-based features, or domain characteristics. [2].

Machine learning (ML) has become a significant weapon in the cybersecurity toolbox in recent years, with the ability to scan large volumes of data and identify patterns suggestive of hostile activity. In this context, ensemble learning—a method that mixes several machine learning models to enhance prediction performance—has gained popularity as a particularly promising strategy. Ensemble

approaches can capture a wider range of features and improve the detection system's overall robustness by utilizing the diversity of different classifiers.[3].
It's important to stress, though, that using these methods maliciously is unethical and may even be against the law. Rather, similar methods ought to be utilized for cybersecurity defense, such developing strong classifiers to identify and lessen the risk of harmful URLs. An effective method for introducing harmful URLs into cybersecurity ecosystems is to use ensemble learning algorithms. Attackers are able to assemble a group that can successfully elude conventional detection systems. First, pertinent attributes including domain reputation, URL structure, and content analysis are retrieved with great care[4] using feature engineering.

II. LITERATURE SURVEY

According to Tsehay Admassu Assegie, phishing poses a lot of issues for the corporate sector. Numerous internet transactions are involved in electronic commerce and banking, including mobile banking. To protect the security of such online transactions, we must distinguish between traits associated with phishing and authentic websites. We gathered information for this study from the public data repository Phish Tank and suggested a K-Nearest Neighbors (KNN) based model for phishing assault detection. The suggested technique uses URL classification to identify phishing attacks. The suggested model's effectiveness is empirically evaluated, and the findings are examined. The test set's experimental results demonstrate the model's effectiveness in detecting phishing attacks.To further improve performance in phishing attack detection, the K value that provides the highest accuracy is identified. [1].
Y. Firdaus has conducted additional study on phishing attempts that target e-commerce websites in particular. To increase detection accuracy, it makes use of machine learning algorithms and features pertaining to the composition and content of online pages. The suggested intelligent solution seeks to offer a strong defense against phishing attacks that is customized to the complexities of online retail transactions by utilizing cutting-edge techniques. By adding machine learning algorithms, the system becomes more adaptive and can identify minute trends that point to phishing activity. Important indications that add to the overall effectiveness of the system are characteristics pertaining to the structure and content of web pages [2].
The quantity of gadgets being connected to the internet has significantly increased in recent years. These gadgets include, but are not restricted to, cloud networks, cellphones, and Internet of Things. Since phishing attacks target human vulnerabilities rather than system flaws, hackers are using them to target these devices more often than other potential cyberattacks. Phishing attacks occur when a user of the internet is tricked into providing personal information, such as credit card numbers or login credentials, by an entity that appears trustworthy. When hackers obtain access to this private information, they can use it as the basis for more complex results. In order to solve this problem, we have created a phishing detection method that only requires nine lexical characteristics in order to identify phishing attempts. Make use of the Naïve Bayes Classifier. For our experiment, we used the ISCXURL-2016 dataset, which contains 11964 instances of both phishing and genuine URLs.[3]
The quantity and variety of web services available on the Web have increased dramatically during the past several years. Online services like social networking, gaming, and banking have quickly advanced, just as people's reliance on them to carry out daily chores has. Consequently, a substantial volume of data is posted to the Internet every day. These online services open up new avenues for interpersonal communication, but they also give crooks new targets. URLs serve as launchpads for all online attacks, allowing a user with bad intentions to send a malicious URL and take the identity of a legitimate person. One of the main components of illicit activity on the Internet is malicious URLs. Due to the risks associated with these websites, regulations requiring defenses to keep end users safe have been established. The suggested method uses a machine learning algorithm called logistic regression to automatically classify URLs into binary categories. [4].

Presenters of this article include D. Annapurna, P. Debnath, K. Sajeevan, and S. Shivangi. i.e., an innovative method utilizing artificial neural networks (ANNs) to identify harmful URLs included in social networking apps. Social media platform expansion has increased the likelihood of coming across harmful URLs, endangering users' privacy and security. The ever-evolving tactics used by fraudsters and the dynamic nature of social media settings provide challenges for traditional methods of URL detection. We suggest using ANNs, which have shown successful in pattern recognition tasks, to automatically detect and categorize harmful URLs in order to address this problem. Using an ANN model trained on a dataset of known benign and dangerous URLs, our method entails extracting pertinent characteristics, using the patterns learned to categorize URLs in real-time within social media applications [5].

Jialong Han, Kai Li, and Shujie Liu: Deep Ensemble learning for malicious url identification. This paper investigates the use of deep ensemble learning methods for malicious URL identification, including ensembles of recurrent and convolutional neural networks (RNNs) and CNNs. According to experimental results, deep ensemble models outperform shallow ensemble models and conventional machine learning techniques in terms of performance. [6]

Arun Kumar, Shubham Jain, and Vivek Kumar. Ensemble of machine learning approaches for malicious URL detection. The usefulness of ensemble learning methods for malicious URL identification, such as bagging, boosting, and stacking, is examined by the authors. They show that ensembles routinely outperform single classifiers in terms of accuracy and robustness when comparing the performance of ensemble models with individual classifiers [7].

An overview of the several machine learning approaches used for malicious URL identification is given in this survey. These approaches include ensemble techniques like bagging, boosting, and stacking. It highlights new developments in the sector and analyzes the advantages and disadvantages of various strategies [8].

A hybrid approach to the problem of phishing attack detection is proposed in another study. A machine-learning model utilizing K-nearest neighbor (KNN) and support vector machine (SVM) is suggested for the automatic detection of phishing attacks. [13] uses the KNN algorithm in conjunction with random forest to detect phishing attacks. A comparison of the KNN and random forest models' performance reveals that random forest outperforms KNN in terms of phishing attack detection [9].

In another study, a model for phishing attack detection is constructed using the logistic regression and Naïve Bayes algorithms. Based on the previous experience that the learning algorithm was given during the training phase, the suggested model is able to identify phishing attacks. When comparing the effectiveness of logistic regression and Naïve Bayes, it can be seen that logistic regression outperformed Naïve Bayes[10].

An automated approach for detecting phishing attacks is provided, utilizing the phishing tank data store and Random Forest technique. The suggested model's effectiveness in identifying suspicious emails is assessed by comparing its performance to the test set's time complexity. The evaluation yields positive results.[11].

A framework for adaptive ensemble learning is presented in Adaptive Ensemble Learning for Malicious URL Detection in order to identify malicious URLs. Based on the properties of incoming URLs, it dynamically modifies the ensemble composition, improving detection performance in real-time circumstances. [12].

Stacking Ensemble in Web Security for the Identification of Malicious URLs.In order to identify dangerous URLs in web security applications, this research presents a stacking ensemble technique. Using a sizable dataset of URLs, the study assesses the performance of a unique architecture for merging various classifiers.[13]

This study looks into the detection of dangerous URLs using stacking ensemble techniques and Gradient Boosting Machines (GBM). The usefulness of the suggested strategy in correctly identifying dangerous URLs while reducing false positives is demonstrated by experimental findings.[14]

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 5, No.11, May : 2024

In order to detect malicious URLs, this research provides an ensemble learning architecture that blends Adaboost and M1 trees. The suggested strategy offers improved detection accuracy and resilience against changing cyberthreats by utilizing the advantages of both algorithms[15].

III. METHODOLOGY

A. Proposed System

We assessed how well categorization methods performed on evolving data using a fraction of a dataset's schemes. This system presents a novel method to improve the prediction accuracy of dangerous URLs by utilizing machine learning techniques.Spambase, MDP-2018, the UCI Phishing website, and balanced and unbalanced phishing datasets were used in the investigation. The MDP 2018 dataset is balanced, in contrast to the imbalanced UCI Phishing website, Spambase datasets, and subset schemes with ratios of 90:10, 80:20, 70:30, and 60:40. The vast number of URLs makes it difficult to reliably identify fraudulent ones. Current approaches might not be able to adequately adjust to the ever-changing strategies used by cybercriminals.This research project's goal is to determine whether a given URL points to a malicious website or not. In order to achieve the best results, the system experiments with different combinations of classification techniques, including Random Forest, Voting Classifier, Stacking Classifier, ABET (AdaBoost.M1 and Extra trees), Gradient Boosting, LightGBM, XG Boost, ROFET (Rotation Forest and Extra trees), BET (Bagging and Extra-trees), Bootstrapping, and combination. However, by investigating more ensemble approaches like Random Forest, XG Boost, LightGBM, and Voting Classifier, we can improve the performance even further. The suggested solution is meant to provide strong defense against constantly changing cyberthreats by utilizing ensemble learning techniques to identify bad URLs. It consists of several parts, such as gathering data, preprocessing, extracting features, training the model, building an ensemble, and evaluating it.

B. System Architecture

A number of essential elements make up the suggested system architecture for identifying malicious URLs, and each one is essential to the detection process. To train and test the models, it starts with the URLs Dataset, which is a set of URLs classified as harmful or benign. The Type of Class URL module controls how URL data is represented

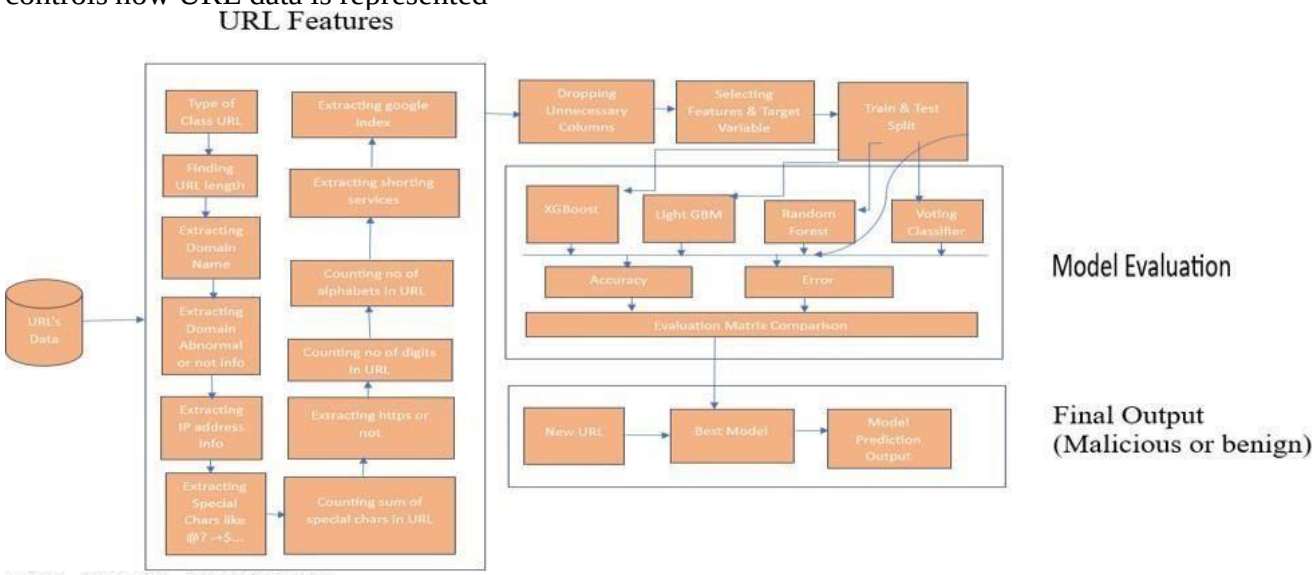

Fig.1 System Architecture

The preprocessed data is used to train ensemble learning models, such as XGBoost, LightGBM, Random Forest, and a Voting Classifier, to provide a variety of classifiers. Evaluation metrics including recall, F1-score, and precision are used to compare these models' performances and choose the best model to predict the output.

(i)URLs Dataset:
This is a representation of the dataset namely MDP 2018. that includes URLs and their labels (Malicious or benign).

(ii)Class URL Type:
Indicates the class or module in charge of representing and managing URL information inside the system.

(iii) Feature extraction modules*:*
These modules extract different features from the URLs, such as the following: URL length; domain name extraction; abnormal domain detection; IP address extraction; special character extraction; extraction from Google index; shortening service detection; counting digits and alphabets; HTTPS detection; sum of special characters.

(iv)Data Preprocessing:
This includes removing irrelevant columns, dealing with null values, and formatting the data appropriately.

(v)Feature Selection:
To enhance model performance, relevant features are chosen from the preprocessed data. Feature selection techniques aims to set of input variables to the ones that are most likely to help a model predict the target variable.

(vi)Train and Test Split: To evaluate the model, divide the preprocessed data into training and testing sets.

(vii)Ensemble Learning Models:
Combining predictions from many base models that includes Voting Classifier, Random Forest, XGBoost LightGBM.

(viii) Evaluation metrics:
Evaluation metrics uses measures like precision, recall, and F1-score to compare the error and accuracy of various models.

(ix) New URL Input:
Indicates newly submitted URLs for the system to classify.

(x) Best Model Selection*:*
This method uses evaluation measures to determine which model performs the best to predict the malicious ones accurately.

(xi) Model Prediction:
Makes use of the chosen model to forecast whether the input URLs will be classified as benign or malicious.

Dataset: The Microsoft Malware Prediction Challenge in
2018 produced the MDP2018 dataset, which includes labeled URLs classified as malicious or benign. Using information taken from URLs and metadata, researchers use it to train and evaluate machine learning models for harmful URL detection. Accuracy, precision, recall, F1-score, and AUC- ROC are examples of evaluation measures. This dataset, which is available for research purposes, helps to progress cybersecurity by making it possible to create strong detection algorithms that counteract changing threats.

	A	B	C
558778	https://www.barrel.net/	benign	0
558779	https://koreantaekwondo.tripod.com/	benign	0
558780	https://taekwondo.wisebytes.net/	benign	0
558781	https://www.steveconway.net/	benign	0
558782	https://www.tkd.net/	benign	0
558783	https://www.taekwondobible.com/	benign	0
558784	https://www.angelfire.com/mi2/540/	benign	0
558785	http://atualizacaodedados.online	malicious	1
558786	http://webmasteradmin.ukit.me/	malicious	1
558787	http://stcdxmt.bigperl.in/klxtv/apps/uk/	malicious	1
558788	https://tubuh-syarikat.com/plugins/fields/files/	malicious	1
558789	http://rolyborgesmd.com/exceword/excel.php?.rand=13InboxLig	malicious	1
558790	http://ongelezen-voda.000webhostapp.com/inloggen.html	malicious	1
558791	http://www.valenzaceramic.com/home/webapps/e52c1/websrc	malicious	1

Fig.2 MDP2018 Dataset

URL Dataset Features

Type of Class URL: The URL class represents Uniform Resource Locators, which let you to manipulate and retrieve online URLs.

Finding URL Length: URL length can be calculated using string manipulation tools such as 'len()'.

Extracting Domain Name: A domain name description includes a website's organization, purpose, and industry focus, which is often seen in the top-level domain, second- level domain, keywords, and brand identification.

Extracting Domain Abnormal or Not Info: "Assessing domain names for anomalies to determine abnormal or typical characteristics."

Extracting IP address info: "Retrieving and analyzing numerical identifiers assigned to network-connected devices to glean location, hosting details, and security implications."

Extracting Special Characters: "Identifying and isolating special characters within a given text or dataset for analysis or processing purposes."

Counting Sum of Special Characters: "Calculating the total number of special characters present within a given text or dataset."

Extracting https or not: "Determining whether a given URL includes the 'https' protocol or not, indicating the presence or absence of secure communication encryption."

Counting no of digits: Counting the digits in a number represented by a URL.

Counting no of Alphabets: Counting the number of letters in a given text represented by a URL.

Extracting shortening services: Use regular expressions to identify shortening services from a given URL.

Extracting google index: Extracting Google search index data via web scraping or Google's Custom Search JSON API.

C. Machine Learning Techniques

Algorithms used in machine learning allow computers to learn from data without the need for explicit programming. On labeled data, supervised learning develops models for classification and regression problems. Unsupervised learning uses dimensionality reduction and clustering to find patterns in unlabeled data. Through interactions with their surroundings and feedback, agents are taught how to make decisions through reinforcement learning. Deep learning excels at tasks like image identification and natural language processing by using neural networks with numerous layers to learn complicated representations. While transfer learning applies knowledge from one domain to another, ensemble learning integrates different models for better performance. These methods find use in a wide range of industries, including marketing, banking, healthcare, and cybersecurity. They also spur innovation and the resolution of challenging issues.

1) Voting Classifier):

A voting classifier is a machine learning model that predicts an output (class) based on the class that has the best chance of becoming the output. It acquires experience by training on a set of several

models. It averages the output class predictions from each classifier that is fed into the voting classifier in order to predict the output class with the biggest majority of votes. Instead of creating distinct, specialized models and assessing each one's correctness, the idea is to create a single model that learns from several models and forecasts output based on the combined majority of votes for each output class. Whether used in a hard voting scheme (mode) or a soft voting scheme (weighted average of projected probabilities), voting classifiers produce predictions that are reliable and frequently more accurate than those made by individual classifiers, particularly when the underlying models have complimentary strengths.

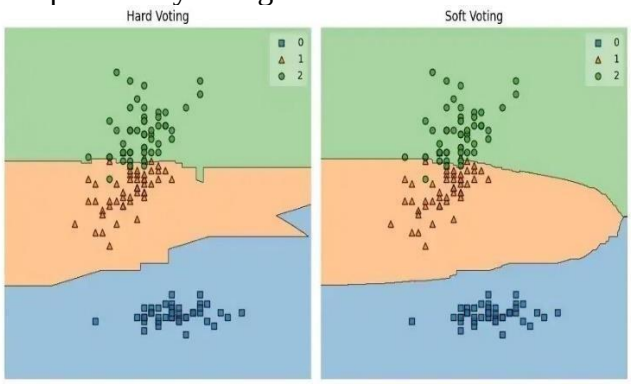

Fig.3 Voting Classifier

a) Hard Voting: A sort of ensemble learning technique called hard voting, or majority voting, combines the predictions from several base models to arrive at a final conclusion. In hard voting, a simple majority vote among the base models determines the final prediction, with each base model in the ensemble independently predicting the class label for a given input.

b) Soft Voting: Soft voting is another ensemble learning technique that combines the predictions from several base models to arrive at a final choice. It is also referred to as weighted voting or probabilistic voting. Soft voting is different from hard voting in that it considers the probability or confidence scores that the base models assign to each class instead of just the class names.

2) Random Forest :

A potent tree learning method in machine learning is the Random Forest algorithm. During the training phase, it generates several Decission Trees. To measure a random subset of characteristics in each partition, a random subset of the data set is used to build each tree. Because each tree is more variable as a result of the randomization, there is less chance of overfitting and overall prediction performance is enhanced. When making predictions, the algorithm averages (for regression tasks) or votes (for classification tasks) the output of each tree. The findings of this cooperative decision- making process, which is aided by the insights of several trees, are consistent and accurate. For regression and classification tasks, random forests are frequently utilized. They are renowned for their capacity to manage complicated data, lessen overfitting, and provide accurate forecasts in many settings

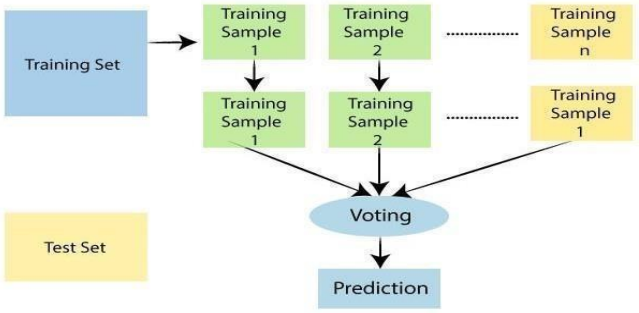

Fig.4 Random Forest Classifier

***a) Ensemble of Decision Trees*:** Random Forest builds an army of Decission trees to take use of the power of ensemble learning. Each of these trees represents a distinct expert with a focus on a certain

area of the data. Crucially, they function separately, reducing the possibility that the subtleties of a single tree will have an undue influence on the model.

b) Random Feature Selection: Random Forest uses random feature selection to make sure every decision tree in the ensemble contributes a different viewpoint. Every tree has a random subset of features selected during training. Because each tree concentrates on a different component of the data due to this randomization, the ensemble as a whole has a diversified range of predictors.

c) Bootstrap Aggregating or Bagging: A key component of Random Forest's training approach is bagging, which is taking numerous bootstrap samples from the original dataset and using them to sample replacement instances. As a result, each decision tree has a different collection of data, which adds diversity to the training process and strengthens the model.

***d) Decision Making and Voting*:** Every decision tree in the Random Forest has an opinion when it comes to predicting outcomes. The mode, or most frequent forecast, across all the trees determines the final prediction for classification tasks. The average of each tree's prediction is calculated in regression tasks. This internal voting system guarantees a collaborative and equitable decision-making process.

3) XG Boost:

A distributed gradient boosting library optimized for efficiency and scalability in machine learning model training is called XGBoost. It is an ensemble learning technique that generates a stronger prediction by aggregating the predictions of several weak models. Extreme Gradient Boosting, or XGBoost, is a machine learning algorithm that has gained popularity and widespread usage because it can handle large datasets and achieve state-of-the-art performance in many machine learning tasks, including regression and classification. XGBoost's effective handling of missing values is one of its primary characteristics, enabling it to handle real-world data with missing values without requiring a lot of pre-processing. Furthermore, XGBoost comes with built-in support for parallel processing, which enables training models on big datasets quickly. Its high degree of customization also enables performance optimization through the fine-tuning of different model parameters.

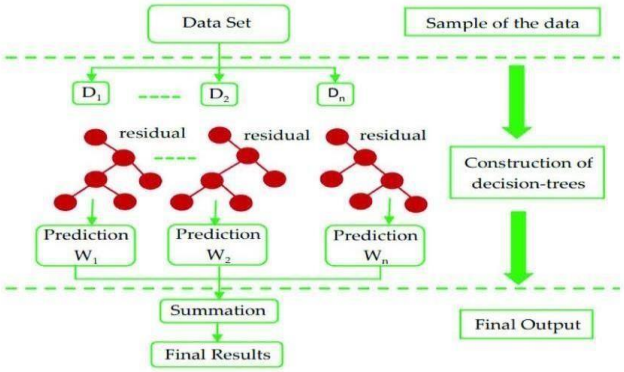

Fig.5 XG Boost Classifier

4) LightGBM:

The Light Gradient Boosting Machine, or LightGBM for short, is a distributed gradient boosting architecture with great performance that performs exceptionally well when working with big datasets and categorical features. Microsoft's LightGBM reduces computational costs by effectively building decision trees leaf-wise using a histogram-based learning technique. This technique works well for big data applications since it is very effective for distributed and parallel computing. Because of its memory efficiency, speed, and capacity to manage unbalanced datasets, LightGBM is a well-liked option for a range of machine learning applications, such as ranking, regression, and classification.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 5, No.11, May : 2024

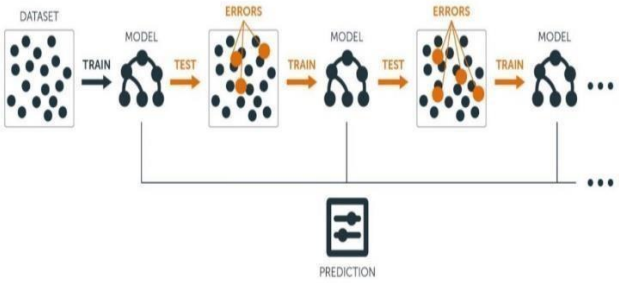

Fig.6 LightGBM Classifier

a) Gradient Boosting Framework: Like other boosting algorithms like XGBoost and Gradient Boosting Machines (GBM), LightGBM is built on the gradient boosting architecture.It progressively constructs an ensemble of weak learners (decision trees), concentrating on the residuals (errors) of the preceding learners in order to build each one.

b) Leaf-wise Tree Growth: LightGBM divides nodes according to the feature that yields the largest reduction in the loss function, and grows the tree level by level using this technique.The leaf-wise approach can result in faster convergence and less memory usage than depth-wise methods, which are employed in conventional gradient boosting algorithms.

c) Gradient-Based Optimization: Like other boosting algorithms, LightGBM uses a gradient-based strategy to maximize the objective function. To direct the process of building a tree, it calculates the gradient of the loss function with respect to the expected values (residuals) for every data point.

d) Histogram-Based Decision Making: In order to accelerate the computation of feature splits during tree construction, LightGBM uses an approach based on histograms.It drastically lowers the computing cost by discretizing the continuous feature values into discrete bins or histograms rather than testing every conceivable threshold for each feature.

e) Gradient-Based One-Side Sampling (GOSS): Gradient- based One-Side Sampling (GOSS) is a technique used by Light GBM to further increase efficiency and decrease overfitting.During the tree-building process, GOSS prioritizes samples that add more to the loss function by sampling data instances with bigger gradients while maintaining the overall data distribution.

f) Exclusive Feature Bundling (EFB): Exclusive Feature Bundling (EFB), a method for minimizing the amount of features by grouping related or correlated characteristics into bundles, is supported by LightGBM. With this feature, the dataset's dimensionality is decreased and computing efficiency is increased without compromising predictive performance.

g) Parallel and Distributed Computing: LightGBM can take advantage of several CPU cores and machines to speed up training on large-scale datasets because it is built for distributed and parallel computing.It is compatible with Hadoop and MPI (Message Passing Interface) distributed computing systems, as well as multi-threading.

IV. RESULTS AND DISCUSSION A) Accuracy

Accuracy is a metric that indicates how frequently a model accurately predicts an outcome. To calculate accuracy, divide the number of correct guesses by the total number of forecasts. The accuracy can be measured on a 0–1 scale or as a percentage. The more accurate, the better. A perfect accuracy of 1.0 is achieved when every prediction made by the model is correct.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 5, No.11, May : 2024

Accuracy = Correct Predictions/All Predictions.

MODEL	ACCURACY
VOTING CLASSIFIER	0.90
RANDOM FOREST	0.90
XG BOOST	0.90
LIGHTGBM	0.90

Fig.7 Evaluation Table

Classification Report:

	precision	recall	f1-score	support
0	0.85	0.95	0.90	93
1	0.95	0.86	0.90	107
accuracy			0.90	200
macro avg	0.90	0.90	0.90	200
weighted avg	0.90	0.90	0.90	200

Fig.8 Model Performance

The model uses the Real time URls to detect the Malicious Ones accurately using ensemble learning techniques. The Figure 8 shows the Model Performance or Evaluation Details(classification report) Evaluation metrices includes precision, recall, f1-Score, support.

B) Confusion Matrix

Confusion matrices are used in machine learning to evaluate the performance of categorization models. This confusion matrix appears to represent the performance of a binary classification model, where the classes are positive and negative. The rows represents the actual classes, and the columns represent expected classes.

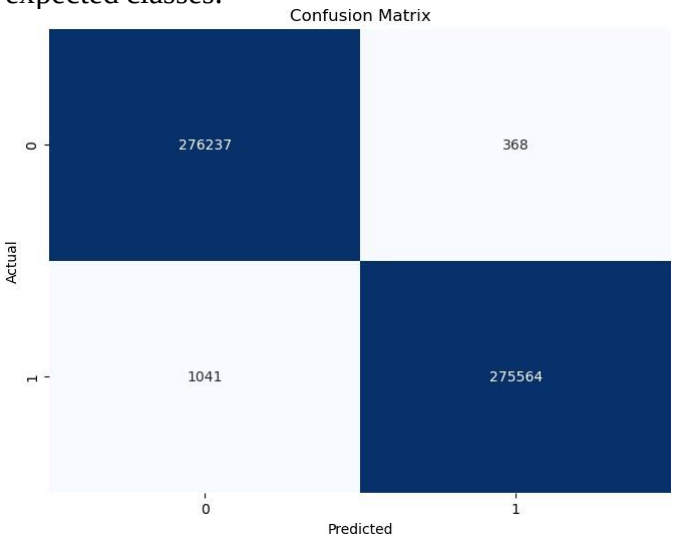

Fig .9 Confusion Matrix

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 5, No.11, May : 2024

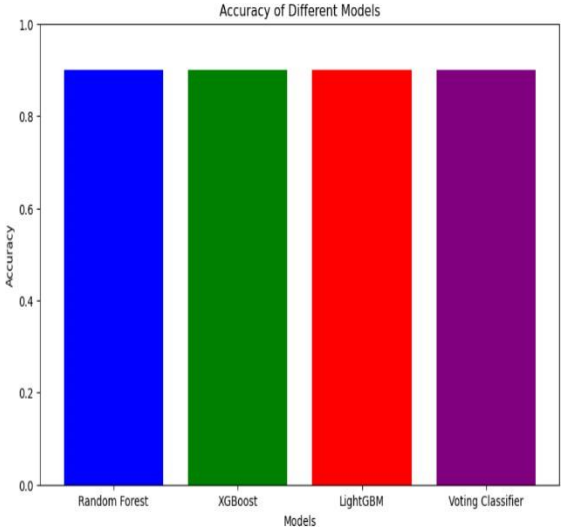

Fig.10 Accuracy of Individual Models

V. CONCLUSION

Ensemble learning algorithms have proven to be a reliable and scalable method for identifying fraudulent URLs, greatly improving cybersecurity protocols. The ensemble model performs better than individual classifiers and conventional detection techniques by combining the predictive powers of several classifiers. By means of thorough assessment, it demonstrates proficiency in identifying various URL attributes suggestive of malevolent intent, thus enhancing precision and adaptability to constantly changing risks.

Continuous monitoring and adaptability to emerging cyber threats are made easier by the ensemble model's deployment and integration in real-world systems. By using a dynamic method, the system is guaranteed to continue protecting users of the internet from any threats related to harmful URLs. In addition, continuous research and development is necessary to improve URL detection systems' usefulness and efficacy and keep them in the forefront of cybersecurity defense. The cybersecurity community can keep developing methods for identifying and thwarting bad online activity by working together and being innovative, which will make the internet a safer place for all users. In conclusion, ensemble learning techniques offer a powerful approach to detecting malicious URLs by leveraging the diversity and complementary nature of multiple classifiers.

REFERENCES

[1] D. Kapil, A. Bansal, N.M.A.J. Anupriya, machine learning based malicious URL detection, **2**, **8**(4S), 22–26 (2020)

[2] H.M.J. Khan, et al., Identifying generic features for malicious url detection system, in 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON) (IEEE, 2019).

[3] H. Kumar, P. Gupta, R.P. Mahapatra, Protocol based ensemble classifier for malicious URL detection, in 2018 3rd International Conference on Contemporary Computing and Informatics (IC3I) (IEEE, 2018).

[4] Kang, H.K.; Shin, S.S.; Kim, D.Y.; Park, S.T. Design and Implementation of Malicious URL Prediction System based on Multiple Machine Learning Algorithms. *J. Korea Multimed. Soc.* **2020**, *23*, 1396– 1405. [**Google Scholar**] [**CrossRef**]

[5] Le, H.; Pham, Q.; Sahoo, D.; Hoi, S.C.H. URLNet: Learning a URL Representation with Deep Learning for Malicious URL Detection. *arXiv preprint* **2018**, arXiv:1802.03162. [**Google Scholar**] [**CrossRef**]

[6] Patil, D.R.; Patil, J.B. Survey on Malicious Web Pages Detection Techniques. *Int. J. u-e-Serv. Sci. Technol.* **2015**, *8*, 195–206. [**Google Scholar**] [**CrossRef**]

[7] Baykara, M.; Gürel, Z.Z. Detection of Phishing Attacks.
In Proceedings of the 2018 6th International Symposium on Digital Forensic and Security (ISDFS), Antalya, Turkey, 22–25 March 2018; pp. 1–5. [**Google Scholar**] [**CrossRef**]
[8] Cova, M.; Kruegel, C.; Vigna, G. Detection and Analysis of Drive-by-Download Attacks and Malicious JavaScript Code. In Proceedings of the 19th International Conference on World Wide Web, Raleigh, NC, USA, 26–30 April 2010; pp. 281–290. [**Google Scholar**] [**CrossRef**]
[9] Singhal, S.; Chawla, U.; Shorey, R. Machine Learning & Concept Drift Based Approach for Malicious Website Detection. In Proceedings of the 2020 International Conference on Communication Systems & Networks (COMSNETS), Bengaluru, India, 7–11 January 2020; pp. 582–585. [**Google Scholar**] [**CrossRef**]
[10] Bhoj, N.; Tripathi, A.; Bisht, G.S.; Dwivedi, A.R.; Pandey, B.; Chhimwal, N. Comparative Analysis of Feature Selection Techniques for Malicious Website Detection in SMOTE Balanced Data. *RS Open J. Innov. Commun. Technol.* **2021**, *2*, 1–10. [**Google Scholar**] [**CrossRef**]
[11] Chaiban, A.; Sovilj, D.; Soliman, H.; Salmon, G.; Lin, X. Investigating the Influence of Feature Sources for Malicious Website Detection. *Appl. Sci.* **2022**, *12*, 2806. [**Google Scholar**] [**CrossRef**]
[12] Altay, B.; Dokeroglu, T.; Cosar, A. Context-Sensitive and Keyword Density-Based Supervised Machine Learning Techniques for Malicious Webpage Detection. *Soft Comput.* **2019**, *23*, 4177–4191. [**Google Scholar**] [**CrossRef**]
[13] Zhuang, W.; Jiang, Q.; Xiong, T. An intelligent anti- phishing strategy model for phishing website detection. In Proceedings of the 2012 32nd International Conference on Distributed Computing Systems Workshops, Macau, China, 18–21 June 2012. [**Google Scholar**] [**CrossRef**]
[14] Invernizzi, L.; Miskovic, S.; Torres, R.; Saha, S.; Lee, S.-J.; Mellia, M.; Kruegel, C.; Vigna, G. Nazca: Detecting Malware Distribution in Large-Scale Networks. *NDSS* **2014**, *14*, 23–26. [**Google Scholar**] [**CrossRef**] [**Green Version**]
[15] Eshete, B.; Kessler, F.B. Effective Analysis, Characterization, and Detection of Malicious Web Pages. In Proceedings of the 22nd International Conference on World Wide Web, Rio de Janeiro, Brazil, 13–17 May 2013; pp. 355–359. [**Google Scholar**] [**CrossRef**]