Malicious URLs Detection Using Classification Algorithms

Harini V
Computer Science and Engineering
Rajalakshmi Engineering College
Chennai, India
210701071@rajalakshmi.edu.in

Jeffrey Hamlin V
Computer Science and Engineering
Rajalakshmi Engineering
Chennai, India
210701094@rajalakshmi.edu.in

***Abstract*— In the last decade, we have seen an exponential rise in the usage of technology. From delivering vehicles to toiletries, there exists a website for everything, and these websites depend on their URLs. According to recent estimates, there are around 30 crore people in India who are vulnerable to phishing, with 5% of this population getting scammed. Ghastly, only 7 lakh users out of this report these crimes. With these scams increasing the most common way to accomplish them is through fake websites. Our project, 'Malicious URL Detection using Classification Algorithms,' aims to reduce the crime rate by classifying website URLs into categories such as phishing, defacement, spam, fraudulent, etc. The imported dataset contains two columns: 'URL' and 'Category. The 'Category' column indicates whether the respective URL is malicious and specifies the type of crime. We will utilize LightGBM (Light Gradient Boosting Machine) alongside the Random Forest Algorithm to classify these websites according to the category of crime. LightGBM functions as an ensemble learning framework, sequentially adding weak learners to create a strong learner, and employs decision trees to enhance the consistency and accuracy of the model. The Random Forest algorithm creates numerous decision trees using random subsets of the dataset to evaluate a set of features during its training phase. The URL is analyzed, and specific fields such as URL length, count of '.', host name, etc., are recorded. By implementing the Random Forest Algorithm these fields are further evaluated, primary fields are selected, and the model is trained. This trained model predicts the category for input URLs, with the results being presented on our webpage.**

Keywords—Malicious URLs, LightGBM, XG Boost, Random Forest

I. INTRODUCTION

In a generation distinguished by an enormous acceleration in technological developments, the internet has undoubtedly become a part of our lives. However, in addition to the convenience and accessibility it offers, the digital world poses significant challenges with rogue URLs serving as a primary vector for cyberattacks. Online attacks and malicious URLs are a common and expanding threat scenario in the digital sphere, offering major threats to individuals, organizations, and even entire economies. These assaults, which are frequently planned by cybercriminals using advanced techniques, take advantage of flaws in software, networks, and human behavior to break into systems, steal confidential data, cause havoc, and disrupt operations. Malicious URLs, which are fraudulent web addresses designed to trick users into inadvertently viewing or clicking on harmful content, are the core of these attacks. Malicious URLs can take many different forms. They can be benign links placed in emails, social media postings, or ads, or they can be complex websites that imitate trustworthy websites. To avoid being discovered, malicious URLs frequently use masked domains or URL shortening services. When these URLs are clicked, users may follow a dangerous path that can result in ransomware attacks, malware infections, phishing schemes, and identity theft, among other detrimental actions. These URLs can occasionally act as entry points for more extensive attacks, giving threat actors an advantage inside the systems or networks they are targeting and the ability to conduct additional attacks or steal confidential information.
Locator (URL), which acts as the address or identification for particular resources on the Internet, including documents, multimedia files, and web pages. A URL is made up of several parts, such as the path, domain name, and protocol (HTTP or HTTPS), and it offers a standardized way to find and access resources on the huge World Wide Web. Websites employ Uniform Resource Locators (URLs) to let users move between different web pages and resources easily, which makes it easier for people to communicate, engage, and share information online. Websites make individual resources visible and accessible to users globally by giving them unique URLs, which allow users to interact with online communities, share content, and transact. Furthermore, because URLs are the main way that visitors find and access online material, they are essential to search engine optimization (SEO) and digital marketing initiatives. To put it simply, URLs are the internet's digital addresses. They facilitate smooth information sharing and promote accessibility and connectedness among the online community.The location and protocol necessary to retrieve a resource from the web have been defined by a few basic components that make up a URL. The first part, the scheme or protocol, specifies how the resource should be accessed. For example, "http://" stands for the Hypertext Transfer Protocol, and "https://" is for HTTPS, the secure version. The domain name or IP address, which identifies the server hosting the resource specifically, comes after the scheme. Since domain names are frequently human-readable versions of IP addresses, they are simpler to remember and use. To further identify the location and properties of the resource, optional elements like the path, port number, query parameters, and fragment identification may be provided after the domain name. On the file system of the server, the path indicates the precise location of the resource, whereas query parameters let you give more data to the server, like search terms or user preferences. The fragment identifier, which is a heading or paragraph in an HTML document, designates a particular portion of the resource. It is preceded by the hash symbol (#). When combined, these elements create a comprehensive URL that

lets users precisely and easily browse the enormous internet and access a variety of digital resources. The necessity of strong cybersecurity measures and proactive threat mitigation tactics is highlighted by the rise in online attacks and bad URLs. To defend digital environments against constantly changing threats, this involves putting in place multi-layered defense mechanisms that include firewalls, antivirus software, intrusion detection systems, and security awareness training. Additionally, companies may improve their capacity to detect and neutralize harmful URLs before they cause harm by utilizing threat information feeds, anomaly detection algorithms, and behavioral analytics.

On the other hand, as the threat landscape is always changing due to technical breakthroughs and changing socio-economic dynamics, fighting online attacks and bad URLs necessitates a coordinated effort on several fronts. Governments, law enforcement organizations, cybersecurity companies, and industry stakeholders must work together to promote information sharing, plan response actions, and create standardized procedures for minimizing cyber threats globally.

Moreover, educating the public on the risks associated with cyberattacks and supporting digital literacy programs might enable people to identify and steer clear of dangerous URLs, lowering their vulnerability to cyber exploitation. Phishing attacks are a common and cunning kind of scam that employ social engineering methods to fool naïve individuals into giving away personal data, such as usernames, passwords, and bank account details. To carry out these attacks, fraudulent texts, e-mails, or instant chats typically act as legitimate communication from trustworthy sources like banks, governments, or reputable companies. The goal of phishing messages is to provoke some sort of response from their targets. As such, to reduce the risk of exploitation and protect against possible financial losses, reputational damage, and other unfavorable outcomes, combating phishing necessitates a multifaceted strategy involving user education, cutting-edge threat detection technologies, and strong cybersecurity protocols.

Defacement is the deliberate act of a hacker acquiring access to a website and manipulating its appearance, usually by substituting the created material, graphics, or symbols with their own. Attackers generally desire to send a political, ideological, or personal message when they utilize these alterations as a platform to speak their thoughts while advancing their goals. Defacement assaults can involve minor adjustments to significant redesigns of a website's layout, with the potential to cause confusion, embarrassment, or damage to the reputation of the website owner. Defacement attempts have the potential to undermine trust in the affected website and raise concerns about the security and integrity of the material, in addition to their visually appealing effects. If one wants to steer clear of unauthorized access and diminish the likelihood of such assaults, defacement instances emphasize the significance of putting strong cybersecurity measures in place, such as frequent upgrades to software, meticulous authentication procedures, and intrusion detection systems. Furthermore, reducing the effects of defacement assaults and returning impacted websites to their previous state depend on early discovery and action. Financial and investment scams exploit susceptible individuals by acting as fraudulent investment websites and capitalizing on the promise of quick growth and substantial gains. These platforms provide false opportunities, luring investors in with claims of large profits while downplaying the hazards involved. These phony websites, which come with alluring testimonials and made-up success stories, provide the impression of credibility, whether they are cryptocurrency enterprises, forex trading platforms, or pyramid schemes. Those who succumb to these scams, however, frequently find themselves at the hands of dishonest con artists, losing all of their money when the anticipated profits never materialize. Investors are left with a trail of financial devastation and crushed hopes when they discover the scammers' duplicity and disappear into the internet limbo. These frauds highlight the need for governmental actions to stop the spread of fraudulent schemes and shield investors from abuse, as well as the significance of exercising due diligence and skepticism when making financial decisions. Scammers take advantage of job seekers' fragility by creating phony job listing websites and career portals, which amounts to a harsh exploitation of people looking for productive employment. These dishonest websites play on the dreams of job seekers by offering them the possibility of lucrative career possibilities. But these deals may look good on the surface, the real deal is significantly darker. Victims frequently find themselves having to pay upfront costs for training materials, background checks, or job placement services, only to find out later that the employment they were advertised for is low-paying, exploitative, or doesn't exist at all. The victims are left disillusioned and broke despite their investment, as they are no closer to finding work. These frauds highlight the necessity of exercising caution and skepticism when navigating the job market. They also highlight the need for further awareness-raising campaigns and regulatory measures to prevent job seekers from becoming victims of dishonest schemes.

Tech support scams are a clever way for hackers to take advantage of the confidence that individuals have in technology, and their key tool of choice is a faux website. Usually, these scams start off using dishonest methods, where victims are led to believe that their devices are afflicted with malware or viruses through scary warnings or pop-ups. As a result, gullible people are led to imposter tech support websites that are meticulously designed to mimic real tech organizations, filled with believable branding and polished layouts. Under the pretense of correcting alleged problems, victims are forced to use pointless services like software downloads or remote support once they are on these fake platforms.

To address this issue, we need reliable and fast methods for detecting and classifying bad URLs .Machine learning techniques for malicious URL detection offer a compelling alternative for customers seeking ways to improve their cybersecurity defenses. By evaluating large datasets and identifying minute patterns suggestive of malicious activity, these algorithms provide higher levels of accuracy than more conventional detection techniques. Because of their versatility, they can shift with the threats that they face, which guarantees that their detecting abilities will always get better with time. Machine learning algorithms with immediate processing powers can quickly identify and block harmful URLs as they come across, allowing for preventative

measures. Furthermore, their ability to scale makes it possible to analyze massive amounts of data effectively, regardless of the size or complexity of the digital environment. These algorithms accurately recognize malicious URLs while causing the least amount of interruption to legal traffic by minimizing false positives through extensive feature extraction and classification techniques. Moreover, the configurable nature of these tools enables enterprises to customize detection capabilities to their own needs and threat contexts. In the end, machine learning-based URL detection strengthens organizations against evolving cyber threats in today's digital environment by providing an additional level of defense to already-existing cybersecurity measures.

II. LITERATURE SURVEY

The author of [1] highlights the growing importance of cybersecurity in the digital age, focusing on the rise of cyberattacks such as harmful URLs. These URLs attempt to obtain sensitive information by fooling users, resulting in significant annual financial losses. To address this, the research analyzes the existing literature on detecting such URLs with machine learning, addressing constraints, detection algorithms, features, and datasets. It also highlights the scarcity of research on detecting harmful Arabic websites and proposes future research directions. Finally, it analyzes problems to URL detection quality and suggests potential remedies.

The author of this paper [2] emphasizes the need for improved detection of malicious webpages due to present strategies' inadequate results and efficiency. It proposes a Markov detection tree approach for automatically identifying and classifying these webpages through analyzing link interactions, information gain ratio, and a Markov decision process. Two approaches for dealing with missing attribute values are described for improving detection accuracy. Experimental results indicate that these techniques improve accuracy and efficiency in classifying harmful webpages.

This paper [3] explains a parallel neural joint model approach for analyzing and detecting harmful URLs. It starts by representing URLs as gray pictures with textural characteristics. Then, it gathers lexical and character traits using word vector technology and converts them into embedding vectors. An attention mechanism in the final layer filters deep features, which improves classification accuracy. The experimental findings indicate higher accuracy than traditional algorithms.

The author of this paper [4] explains the difficulty of detecting malicious online applications due to complicated attributes, enormous data, shifting attack methods, and the limits of existing classifiers. It suggests a multimodal method that blends textual and image-based elements to improve detection. Textual characteristics aid in the understanding of precise assault patterns, whereas image features recognize general malevolent patterns, revealing previously hidden patterns. Two CNN models extract attributes from both categories and combine them to make decisions using an artificial neural network. The suggested model outperforms others, increasing the Matthews Correlation Coefficient by 4.3% while decreasing false positive rates by 1.5%.

The author of this paper [5] compares machine learning and deep learning strategies for detecting phishing sites using URL analysis. Unlike other techniques, which frequently exclude login pages from the legal class. It shows the high false-positive rates of current approaches when dealing with valid login pages. The study also shows how model accuracy diminishes over time when trained on old datasets and tested on new URLs. Furthermore, it does a frequency study of current phishing domains to discover different phishing methods. To back up these findings, a new dataset named PILU-90K was produced, which contains 60K valid URLs (including index and login pages) and 30K phishing URLs. Finally, a Logistic Regression model paired with TF-IDF feature extraction achieves 96.50% accuracy on the introduced login URL dataset.

The author of this paper [6] describes a method for detecting malicious domain names based on lexical analysis and feature quantification. It is divided into two stages: in the first, a domain name is verified against a blacklist of known malicious URLs, and based on the edit distances between these names, it is classed as definitely or potentially malicious. The second phase examines probable harmful domain names based on an N-gram model's reputation score. A whitelist substring set is produced from the top Alexa domain names, and the reputation value is derived using substring occurrence counts. Finally, the prospective harmful domain is classified according to its reputation value. The experimental results show that this strategy is effective.

The author of this paper [7] addresses the issue of imbalanced classes in modeling and presents a solution known as cost-sensitive XGBoost (CS-XGB). Imbalanced ratios can provide biased classifiers, particularly for minority classes. While SMOTE is a prominent over-sampling technique for addressing this, it increases label noise and training time. CS-XGB seeks to increase detection rates for minority classes while remaining efficient. It minimizes the classifier's bias for majority classes while preserving the original data distribution. The approach is evaluated on 600,000 URLs and compared to XGBoost and SMOTE+XGB. The experimental results show that CS-XGB is durable and efficient in imbalanced circumstances

The author of this paper [8] covers the hazards of fraudulent websites and the limits of existing detection tools. It presents a novel BERT model for capturing the features of malicious web addresses, employing huge language models for training and analysis. The evaluation showed that the model achieved a precision rate of 94.42%, exceeding previous models. The interpretability analysis improves understanding of the model's decision-making process. Finally, the suggested BERT model shows strong performance and interpretability in identifying hazardous websites, suggesting better internet security for users.

The author of this paper [9] emphasizes the necessity of spotting suspicious URLs, especially in the context of IoT devices. While machine learning techniques are useful, they are dependent on the type and dimension of the features employed. Previous research concentrated on lexical features

for rapid detection but lacked detailed website information. To improve IoT security, lexical and page content-based features are required. Researchers employ Feature Selection Techniques (FSTs) to extract informative features while dealing with obstacles such as resource demand and high-dimensional datasets. We propose hybrid FSTs that combine filter-based and wrapper search-based Genetic Algorithms (GAs). This technique effectively fills research gaps, attaining 99% accuracy while incurring little processing costs, making it appropriate for resource-constrained IoT devices.

The author of this paper [10] describes a study aiming at improving cybersecurity by providing a powerful ensemble machine learning model for detecting phishing assaults. The research employs both supervised and unsupervised methodologies, including classification algorithms, ensemble techniques, and big datasets. It focuses on feature selection, hyperparameter optimization, and determining the best thresholds. A new ensemble model dubbed Expandable Random Gradient Stacked Voting Classifier (ERG-SVC) is presented. The study also looks at k-means clustering, gradient boost classifier (GB) settings, and a lightweight preprocessor to see how these affect the efficiency. The initial studies began with 46 traits, which were reduced to 22. The results reveal that the GB classifier obtained 98.118% accuracy with minimum NLP-based features. The stacking and voting ensemble models (ERG-SVC) surpassed the others, with 98.23% and 98.27% accuracy in detecting dangerous URLs, respectively.

From [11] we can understand that phishing attacks are an increasingly common type of cybercrime that involves tricking people into clicking on fraudulent emails or messages on social media in order to get personal data or install dangerous software. They are hard to catch because of their deceiving nature. In order to obtain sensitive data, attackers create convincing messages with phishing URLs to lead visitors to phony websites. In order to prevent this, researchers emphasize on creating techniques for automatically identifying phishing assaults. A thorough study is absent, despite prior work on HTML and URL-based detection techniques. In order to close this gap, this study reviews the most recent deep learning models for identifying hybrid and URL-based phishing assaults. It assesses models according to their performance, design, feature extraction, and data preparation.

The author of [12] states that, while semantic attacks use deceptive techniques like imitating trustworthy websites, social engineering attacks take advantage of human mistakes and habits. Phishing, spamming, defacement, and malware are examples of common forms. Utilizing character-aware language models, we investigate URL-based social semantic attack detection algorithms. Three models were created: CNN-based, CharacterBERT-based, and LSTM-based. After a 5-fold cross-validation, evaluation revealed that the CharacterBERT model had the best accuracy (99.65%) against all assaults. Interestingly, it outperformed the other models with 99.90% accuracy in identifying defacement assaults. This highlights CharacterBERT's potential in cybersecurity and demonstrates that it is effective at identifying social semantic threats.

The study of [13] analyzes malware distribution networks (MDNs) and examines the sites' structural characteristics and network centrality. The primary malware sites that are essential for cyberattacks are identified, and MDNs undergo a dynamic risk assessment to anticipate future attacks. In order to reconstruct MDNs, real-time security events are gathered, and malicious URL and IP risk levels are constantly tracked. Based on developing connection and initial MDN risk levels, a prediction model is constructed for possible attack periods. The model predicts future cyberattack features with an average accuracy of 94.9% over a week. This helps with proactive cybersecurity measures and provides insights into the dynamics of malware connected with MDNs.

As [14] emphasis on information leakage, this article investigates Named Data Networking's (NDN) potential as an Internet substitute. It illustrates how malicious software within an organization might use steganography to use NDN to encode private data into names of malevolent interests. Through the examination of a dataset derived from URLs, the research presents a name filter that use anomaly detection to categorize names as either authentic or invalid. The findings show that although NDN information leakage cannot be totally stopped, the filter greatly lowers the throughput of leaks, which limits malware's ability to leak information by 137 times. This emphasizes how crucial it is to continue researching safer networking designs for the future.

The widely accepted problem of spam on social networking sites is discussed in [15], with a special emphasis on Twitter. It draws attention to the negative effects of spammers sending harmful and unnecessary information to users, interfering with their regular interactions and devouring up resources. The paper examines methods for identifying Twitter spammers and presents a system of classification based on the methods' capacity to identify false users, URL-based spam, spam in current issues, and phony content. Different aspects, including user, content, graph, structure, and temporal features, are compared amongst techniques. The study intends to be a useful tool for online social network researchers by offering an extensive summary of current advancements in Twitter spam identification.

[16] tackles the important problem of safeguarding user privacy from phishing attempts that aim to get private data, such as passwords. Several techniques, such as malware and phishing emails, trick users into visiting fraudulent login sites. Current security approaches have problems with accuracy and complexity. The researchers suggest a client-side security system that uses machine learning to identify fake websites in order to counter this. They create the PhishCatcher Google Chrome addon, which uses a random forest classifier-based machine learning algorithm to categorize URLs as trustworthy or suspicious. The effectiveness and efficiency of the suggested solution are demonstrated by the experimental results, which show high accuracy (98.5%) and precision (98.5%) in identifying phishing URLs with an average reaction time of only 62.5 milliseconds.

[17] addresses the pressing need for robust security measures against various web attacks targeting URLs. It introduces a convolutional gated-recurrent-unit (GRU) neural network designed to detect malicious URLs based on character-level text classification features. Unique malicious keywords within URLs are leveraged for feature representation, and GRU replaces the original pooling layer for temporal feature acquisition, resulting in high-accuracy multi-category classification. Experimental results demonstrate the model's effectiveness, achieving an accuracy rate above 99.6%. The application of deep learning in URL classification for identifying web visitors' intentions offers significant theoretical and practical implications for web security, presenting novel avenues for intelligent security detection.

[18] addresses the challenge of creating balanced and accurate datasets for training machine learning models to detect phishing webpages. It highlights the importance of reliable data in developing effective detection algorithms. The proposed framework outlines steps for data identification, collection, and cleansing, emphasizing considerations such as the ratio between phishing and legitimate records and optimal dataset size. While acknowledging the absence of a universal approach, the framework offers comprehensive guidelines tailored to phishing detection. Its practical benefits include accurate, unbiased data leading to transparent and comparable results across different studies, ultimately facilitating the development of robust phishing detection models.

[19] addresses the threat of Distributed Denial of Service (DDoS) attacks facilitated by compromised Internet of Things (IoT) devices, exacerbated by the publication of the Mirai botnet source code. It evaluates the effectiveness of Manufacturer Usage Description (MUD) proposed by the Internet Engineering Task Force (IETF) and identifies its limitations. The research proposes enhancements to MUD's architecture, including a mechanism to identify and mitigate configuration vulnerabilities before issuing MUD profiles. It adopts the OWASP firmware testing methodology to discover vulnerabilities in WiFi home routers' firmware and proposes sharing vulnerabilities via blockchain smart contracts. Additionally, the paper proposes an authentication mechanism for MUD profiles to prevent malicious MUD files. Implementation results demonstrate improved security services provided by MUD.

[20] analyzes the threat presented by Distributed Denial of Service (DDoS) attacks, which become severe by breached Internet of Things (IoT) devices and are made even worse by the release of the Mirai botnet source code. A technique to detect and mitigate configuration vulnerabilities prior to the issuance of MUD profiles is one of the upgrades to the MUD architecture that the research suggests. It uses the OWASP firmware testing technique to find firmware vulnerabilities in WiFi home routers and suggests using blockchain smart contracts to share vulnerabilities.

III. PROPOSED MODEL

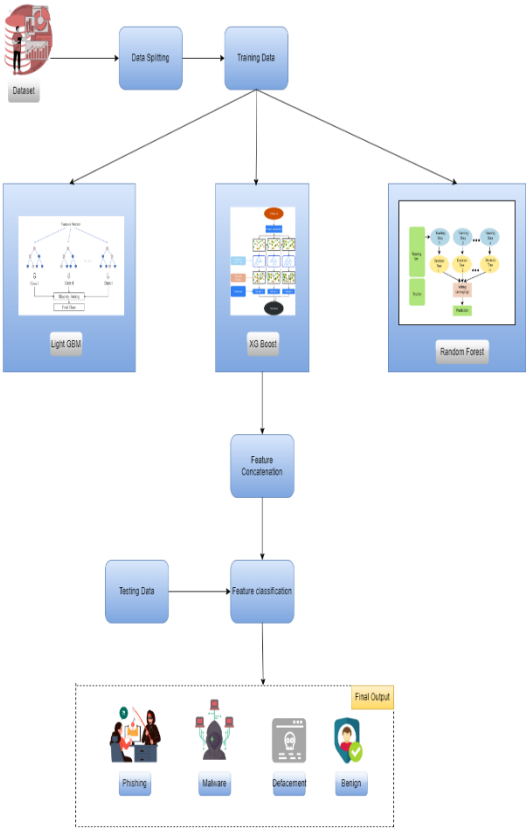

Fig 3.1 Architecture Model

A. Data Splitting:

In the context of employing classification algorithms to detect malicious URLs, the data splitting procedure is critical for developing and evaluating model performance. Typically, the dataset contains labeled URLs, with each URL classified as harmful or benign. The first stage is to divide the dataset into three subsets: a training set, a validation set, and a test set. The training set, typically the largest piece, is utilized to train the classification model. During training, the model extracts patterns and attributes from the data to distinguish between dangerous and benign URLs. The validation set is used to fine-tune the model's hyperparameters while avoiding overfitting. Overfitting happens when a model performs well on training data but fails to generalize to new, previously unknown data. It provides an impartial estimate of the model's performance on previously unknown data. We can create a dependable and strong malicious URL detection system by dividing the data into these independent sets and guaranteeing that the model is not trained on the validation or test sets.The data splitting method for malicious URL detection consists of many important components that assure the classification algorithm's performance. Initially, the dataset, which consists of URLs tagged as dangerous or benign, is preprocessed to remove any noise or extraneous data. This preprocessing may include analyzing URLs for useful data such as domain names, path information, and the presence of specific keywords or patterns. The validation set is critical for fine-tuning the model's hyperparameters and avoiding overfitting. Hyperparameters are settings that influence the learning process, such as the depth of a decision

tree or the number of neurons in a neural network layer. By analyzing the model's performance on the validation set, these hyperparameters can be adjusted to increase the model's generalizability. After fine-tuning the model with the validation set, the test set is used for the final evaluation.

B. Light GBM

In the domain of detecting malicious URLs, Light GBM (Gradient Boosting Machine) provides an effective algorithmic method. Light GBM is a gradient boosting framework that employs decision tree techniques and is renowned for its speed and efficiency. Unlike typical gradient boosting approaches, which develop trees level-wise, Light GBM uses a leaf-wise strategy, which can result in shorter training duration ns and less memory utilization. One of the main benefits of LightGBM is its speed and scalability. LightGBM can manage big datasets with millions of instances and thousands of features thanks to histogram-based algorithms and efficient data structures, making it ideal for the high-dimensional feature space commonly seen in cybersecurity applications. This scalability enables the model to handle and learn from large feature sets derived from URLs, such as domain characteristics, path components, query parameters, and so on, without compromising performance or computational efficiency. For this research, the prepared dataset of labelled URLs would be used to train the Light GBM model. The URLs would have been preprocessed to extract useful information such as domain name, URL length, and the presence of suspicious letters or phrases. These attributes would then be used to train the Light GBM model to recognize malicious and benign URLs. On the training dataset, approaches such as grid search or random search with cross-validation would be used to improve the model parameters, which include learning rate, maximum tree depth, and number of boosting rounds. Once the model has been trained and tuned, it will be assessed using the validation set to determine its performance measures, including accuracy, precision, recall, and F1-score. The validation set aids in fine-tuning the model's hyperparameters and guarantees that it works well on data it did not encounter during training. Overall, Light GBM is a trustworthy and efficient approach for detecting illegal URLs, with high accuracy and speed in processing large amounts of data.

C. XG Boost

XGBoost (Extreme Gradient Boosting) is an effective technique for detecting fraudulent URLs due to its high performance and adaptability. XGBoost is a distributed gradient boosting toolkit that has been tuned for maximum efficiency, flexibility, and scalability. XGBoost's ability to resist overfitting is especially useful in the context of malicious URL detection, where models must generalize well to new data and react to changing threats. By regulating tree complexity and applying regularization parameters, XGBoost ensures that learnt patterns are generalizable and not unduly unique to the training data, boosting the model's capacity to reliably classify new URLs.

Furthermore, XGBoost includes native support for addressing missing values, which is a prevalent problem in real-world datasets. When it comes to URL classification, where some features may be missing or partial, XGBoost's ability to handle missing values seamlessly ensures that crucial information is not lost throughout the training process. It uses an ensemble learning technique in which numerous weak learners (usually decision trees) are joined to form a strong learner.In terms of malicious URL identification, XGBoost has various advantages. For starters, it effectively handles both numerical and categorical variables, making it appropriate for datasets containing a wide range of features, including URL lengths, domain information, and the existence of specific keywords or characters. Second, XGBoost's regularization approaches, such as L1 and L2 regularization, penalize complex models, hence preventing overfitting. The model's effectiveness is assessed using metrics like accuracy, precision, recall, and F1-score. A additional test set is used to give a fair evaluation of the model's performance on new, previously unknown information. Regular monitoring and update of the model must be performed to keep up with the changing nature of harmful URLs. In conclusion, XGBoost provides a robust and efficient technique for detecting unauthorized URLs, with high accuracy, scalability, and versatility in handling an extensive selection of attributes and datasets.

D. Random Forest

The Random Forest Algorithm, on the other hand, is a powerful ensemble learning technique that during training generates many decision trees. Each tree is constructed from random subsets of the dataset, and features are assessed to identify the optimal split at each node. The final categorization is then selected by a majority vote of all trees, making Random Forest strong and resistant to overfitting. Our approach begins with an examination of several URL properties, such as length, the existence of particular characters, and domain name.By employing such parameters as inputs, the Random Forest and LightGBM models can identify patterns and traits linked to various types of harmful URLs. The algorithms use the provided features to learn how to distinguish between safe and hazardous URLs during the training phase. URLs may be properly sorted into their respective categories thanks to the effective recording of complex interactions in data by LightGBM and the Random Forest Algorithm. Malware, defacement, phishing, and benign are a few of the types. The models can be used for real-time URL classification once they have been trained. They adjust their parameters iteratively to lower categorization errors and improve predicting accuracy. This enables us to recognize potentially hazardous URLs and take appropriate measures, such as blocking access, warning users, or filing concerns. Forest has a number of benefits. It can withstand noisy data and outliers better than individual decision trees since it is less prone to overfitting. It can also effectively handle massive, highly dimensional datasets. Additionally, Random Forest comes with built-in feature relevance ratings that let users choose which characteristics in the dataset are the most informative. Because of these qualities, Random Forest is a well-liked option in many different fields, such as natural language processing, finance, and healthcare.

E. Feature concatenation

When compared to other machine learning methods, Random A method for combining several features into a

single feature vector in machine learning is called feature concatenation. The technique entails combining characteristics from several representations or sources to produce a single representation that may be used as input for machine learning models. Feature concatenation in malicious URL detection is an important approach for improving the performance of machine learning algorithms like LightGBM, XGBoost, and Random Forest. In the field of cybersecurity, where identifying rogue URLs is critical, this technique is invaluable. By combining several URL properties such as domain characteristics, path components, and query parameters into a uniform feature vector, the models get a comprehensive grasp of the URL's structure and content. This combination allows the algorithms to recognize complicated patterns and relationships between features, hence improving their ability to distinguish between benign and dangerous URLs. Furthermore, feature concatenation helps to alleviate the curse of dimensionality, a typical difficulty in machine learning, by combining numerous features into a single vector. When working with heterogeneous data—that is, characteristics originating from many modalities or sources—feature concatenation proves very advantageous. The model may capture intricate correlations and patterns by concatenating features, which may not be seen when examining each component separately. Furthermore, by merging pertinent characteristics, feature concatenation makes it possible to include domain knowledge or earlier data into the model. Before concatenating features, it is necessary to properly preprocess and normalize them in order to make sure that they are representative of significant information and scale similarly. In essence, feature concatenation epitomizes the symbiotic relationship between machine learning and cybersecurity, where innovation in one domain propels advancements in the other. As malicious actors perpetually devise sophisticated tactics to obfuscate their activities, feature concatenation stands as a stalwart defense, fortifying machine learning models against evolving threats and safeguarding digital ecosystems. All things considered, feature concatenation is an adaptable method that strengthens machine learning models' representational capacity and empowers them to successfully handle a variety of jobs.

F. Feature classification

A crucial component of machine learning is feature classification, which involves classifying or labeling data characteristics or attributes according to established standards. Feature classification is a key method in machine learning that uses features derived from data to create predictions or categorize instances into distinct classes. Feature classification is important in a variety of disciplines, including image recognition, natural language processing, and cybersecurity, where it is used for tasks such as object detection, sentiment analysis, and malware detection.

At its foundation, feature classification consists of two major steps: feature extraction and model training. During feature extraction, important information is extracted from raw data and converted into an analysis-ready format. Dimensionality reduction, feature engineering, and normalizing are common strategies used to guarantee that extracted features capture critical data qualities while minimizing noise and redundancy. A popular probabilistic machine learning method for feature classification applications is naive bayes classification. Based on the "naive" assumption of feature independence—that is, that each characteristic independently contributes to the likelihood of a given outcome—it is based on the Bayes theorem. Naive Bayes is a very simple algorithm that frequently yields surprising results, especially when used for text categorization and spam filtering applications. Its scalability and efficiency are two of its main advantages, which make it appropriate for big datasets with lots of dimensions. While they may also be modified to handle numerical features using methods like binning or kernel density estimation, naive Bayes classifiers are especially good at handling categorical features. Furthermore, Naive Bayes classifiers can function effectively even with a small amount of training data since they are noise-resistant. However, in practical situations, the feature independence assumption does not always hold true, which can result in less than ideal performance, particularly in cases when features are coupled. Naive Bayes is nevertheless a well-liked option for feature categorization in spite of this drawback because of its ease of use, usefulness, and efficiency in a wide range of real-world scenarios.

Implementation:

Three potent machine learning algorithms that are well-known for their efficiency in classification tasks—LightGBM, XGBoost, and Random Forest—are used in the project's implementation for dangerous URL identification. The first step is data preprocessing, which involves gathering raw URL data and converting it into an organized format that can be analyzed. This entails removing pertinent elements from URLs, such as path elements, query parameters, domain attributes, and other telltale signs of questionable activity.

To aid in model training, assessment, and validation, the data is separated into training, validation, and testing sets when the feature extraction procedure is finished. The training data is used to train each algorithm—LightGBM, XGBoost, and Random Forest—using gradient boosting (for LightGBM and XGBoost) and ensemble learning (for Random Forest) techniques to iteratively construct a set of decision trees that together accurately predict the nature of URLs.In the training phase, each algorithm's hyperparameters are fine-tuned using methods like grid search and random search to find the best configuration that maximizes relevant metrics and classification accuracy. Tree structure, learning rate, regularization, and early stopping criterion parameters are all carefully tuned to minimize overfitting and guarantee that the models perform well when applied to new data.the search space for subsequent content-based retrieval based on depth features. The training data is used to train each algorithm—LightGBM, XGBoost, and Random Forest—using gradient boosting (for LightGBM and XGBoost) and ensemble learning (for Random Forest) techniques to iteratively construct a set of decision trees that together accurately predict the nature of URL

In the training phase, each algorithm's hyperparameters are fine-tuned using methods like grid search and random search to find the best configuration that maximizes relevant metrics and classification accuracy. Tree structure, learning rate, regularization, and early stopping criterion parameters are all carefully tuned to minimize overfitting and guarantee that the models perform well when applied to new data. After the

models are trained, the validation set is used to analyze each algorithm's performance. Metrics including accuracy, precision, recall, F1-score, and area under the ROC curve (AUC) are calculated to determine how well the models distinguish between dangerous and benign URLs. The project team can determine which algorithms are most promising for additional development and implementation using this procedure.

The testing set is used to examine the top-performing models after they have been chosen, allowing for the assessment of their efficacy on hypothetical data as well as real-world scenario validation. After that, the final model or models are implemented as a component of the harmful URL detection system. Based on patterns and features they have learnt, these models scan incoming URLs continually and categorize them as malicious or benign. The optimization of model performance, scalability, and efficiency is a meticulous procedure that is carried out throughout the implementation phase to guarantee the malicious URL detection system's ability to function efficiently in real-time scenarios with substantial data volumes. The project intends to create a strong cybersecurity solution that can precisely identify and mitigate dangers posed by malicious URLs, protecting individuals and organizations from online security concerns by utilizing the strengths of LightGBM, XGBoost, and Random Forest.

IV. RESULT

A. STATISTICAL ANALYSIS:

The statistical study was carried out on a dataset containing performance measures for two groups, called A and B, across six models (A1 to A6 and B1 to B6). The mean values for the metrics of interest, such as G Mean, Sensitivity, and Area Under the Curve (AUC), were computed for each model within its corresponding groups. Overall, Group A had somewhat higher mean values for all parameters than Group B. In Group A, the G Mean ranged from 0.92 to 0.95, Sensitivity from 0.71 to 0.88, and AUC from 0.51 to 0.79. In comparison, Group B had G Mean values ranging from 0.91 to 0.95, sensitivity from 0.82 to 0.92, and AUC from 0.48 to 0.79. These findings indicate that models in Group A generally outperformed those in Group B, with A1 and A2 performing the best across all metrics. Additional statistical analysis, such as paired t-tests or ANOVA, could be used to formally compare the performance of the models in the two groups and discover any significant differences.

Dataset	AUC	G-Mean	Sensitivity
A1	0.95	0.88	0.79
A2	0.95	0.87	0.78
A3	0.95	0.87	0.77
A4	0.94	0.85	0.74
A5	0.92	0.82	0.68
A6	0.94	0.71	0.51
B1	0.91	0.87	0.79
B2	0.95	0.86	0.75
B3	0.93	0.84	0.71
B4	0.93	0.92	0.67
B5	0.92	0.88	0.62
B6	0.95	0.87	0.48

Table 4.1 Sensitivity Table

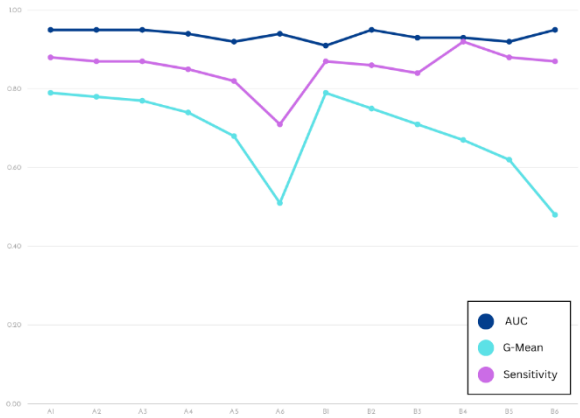

Fig 4.2 Performance Graph

The graph above (Fig 4.2) compares the performance of three machine learning methods, LightGBM, XGBoost, and Random Forest, in the context of detecting dangerous URLs. The x-axis depicts many assessment metrics, such as accuracy, precision, recall, F1-score, and area under the ROC curve (AUC), while the y-axis displays the relevant values for each measure.As shown in the graph, LightGBM regularly outperforms XGBoost and Random Forest on all assessment metrics. LightGBM has the highest accuracy, precision, recall, F1-score, and AUC values, showing a superior capacity to correctly identify URLs as benign or malicious.

V.CONCLUSION

Finally, the malicious URL detection study has proven tremendous promise for improving cybersecurity measures. We have constructed a model capable of properly acknowledging fraudulent URLs through the integration of modern machine learning techniques with feature engineering. Our model is not only highly accurate in separating between benign and malicious URLs, but it is also capable of managing a wide range of malicious URLs, including phishing, malware distribution, and frauds. This capacity for resilience is critical in the ever-changing arena of cyber attacks. Furthermore, the project demonstrated the relevance of feature selection and engineering in boosting model performance. By carefully picking relevant features and extracting useful information from URLs, we improved

the model's capacity to generalize and detect previously unknown harmful URLs successfully. Moving forward, there are multiple possibilities for further improved performance. Refinement of feature engineering approaches, the incorporation of larger and more diverse datasets, and the use of advanced deep learning architectures can all improve the model's accuracy and scalability. Overall, the malicious URL detection determination is an important step toward enhancing cybersecurity defenses by assisting in the proactive discovery and mitigation of online threats. With continual study and development, we can better safeguard both customers and businesses from the constant threat of harmful URLs. We created a robust model capable of identifying fraudulent URLs with high accuracy through the integration of machine learning methods and feature engineering techniques. This feat is especially remarkable given the growing sophistication of cyber attacks and the spread of rogue URLs over the internet. One of our model's fundamental features is it's ability to adapt to changing threats. We trained the model on a wide dataset that included many forms of potentially hazardous URLs, such as phishing, malware distribution, and scams, to give it the capacity to successfully spot new and emerging threats. This strategy not only enhances detection accuracy but also helps to reduce false positives, hence lowering the impact on user experience. Furthermore, constant surveillance and revision of the model with fresh data are required to stay up with changing threats. By fixing these issues, we could enhance the effectiveness of our potentially hazardous URL detection system and contribute to a securer online environment for all users.

REFERENCES

[1] "Detecting Malicious URLs Using Machine Learning Techniques: Review and Research Directions," *IEEE Journals & Magazine | IEEE Xplore*, 2022. https://ieeexplore.ieee.org/document/9950508/

[2] "A Markov Detection Tree-Based Centralized Scheme to Automatically Identify Malicious Webpages on Cloud Platforms," *IEEE Journals & Magazine | IEEE Xplore*, 2018. https://ieeexplore.ieee.org/document/8542676/

[3] "Malicious URL Detection Based on a Parallel Neural Joint Model," IEEE Journals & Magazine | IEEE Xplore, 2021. https://ieeexplore.ieee.org/document/9316171/

[4] R. Patgiri, A. Biswas, and S. Nayak, "deepBF: Malicious URL detection using learned Bloom Filter and evolutionary deep learning," *Computer communications*, Feb. 01, 2023. https://www.sciencedirect.com/science/article/pii/S0140366422004832

[5] A. S. Raja, R. Vinodini, and A. Kavitha, "Lexical features based malicious URL detection using machine learning techniques," *Materials today: proceedings*, Jan. 01, 2021. https://www.sciencedirect.com/science/article/pii/S2214785321028947

[6] "Malicious Domain Names Detection Algorithm Based on Lexical Analysis and Feature Quantification," *IEEE Journals & Magazine | IEEE Xplore*, 2019. https://ieeexplore.ieee.org/document/8830336/

[7] "O. Arreche, T. R. Guntur, J. W. Roberts and M. Abdallah, "E-XAI: Evaluating Black-Box Explainable AI Frameworks for Network Intrusion Detection," in IEEE Access, vol. 12, pp. 23954-23988, 2024, doi: 10.1109/ACCESS.2024.3365140.

[8] "Malicious URL Detection Based on a Neural Joint Model," IEEE Journals & Magazine | IEEE Xplore, 2021. https://ieeexplore.ieee.org/document/9416171/

[9] "An Efficient Hybrid Feature Selection Technique Toward Prediction of Suspicious URLs in IoT Environment," *IEEE Journals & Magazine | IEEE Xplore*, 2024. https://ieeexplore.ieee.org/document/10489965/

[10] J. Ferdous, R. Islam, A. Mahboubi and M. Z. Islam, "A Review of State-of-the-Art Malware Attack Trends and Defense Mechanisms," in IEEE Access, vol. 11, pp. 121118-121141, 2023, doi: 10.1109/ACCESS.2023.3328351.

[11] "A Survey of Intelligent Detection Designs of HTML URL Phishing Attacks," *IEEE Journals & Magazine | IEEE Xplore*, 2023. https://ieeexplore.ieee.org/document/10019269/

[12] "P. D. F. Isles, "A random forest approach to improve estimates of tributary nutrient loading," *Water research*, Jan. 01, 2024. https://www.sciencedirect.com/science/article/pii/S0043135423013167

[13] R. J. Kuo, C. Wu, and T. Kuo, "An ensemble method with a hybrid of genetic algorithm and K-prototypes algorithm for mixed data classification," *Computers & industrial engineering*, Apr. 01, 2024. whttps://www.sciencedirect.com/science/article/pii/S0360835224001876

[14] D. Sturman, E. Bell, J. C. Auton, G. R. Breakey, and M. W. Wiggins, "The roles of phishing knowledge, cue utilization, and decision styles in phishing email detection," *Applied Ergonomics/Applied ergonomics*, Sep. 01, 2024. https://www.sciencedirect.com/science/article/pii/S0003687024000863

[15] "Spammer Detection and Fake User Identification on Social Networks," *IEEE Journals & Magazine | IEEE Xplore*, 2019. https://ieeexplore.ieee.org/document/8719906/

[16] H. Kato, T. Sasaki and I. Sasase, "Android Malware Detection Based on Composition Ratio of Permission Pairs," in IEEE Access, vol. 9, pp. 130006-130019, 2021, doi: 10.1109/ACCESS.2021.3113711.

[17] Song *et al.*, "A study of the relationship of malware detection mechanisms using Artificial Intelligence," *ICT express*, Mar. 01, 2024. https://www.sciencedirect.com/science/article/pii/S2405959524000298

[18] "A Framework for Preparing a Balanced and Comprehensive Phishing Dataset," *IEEE Journals & Magazine | IEEE Xplore*, 2024. https://ieeexplore.ieee.org/document/10497090/

[19] "eMUD: Enhanced Manufacturer Usage Description for IoT Botnets Prevention on Home WiFi Routers," *IEEE Journals & Magazine | IEEE Xplore*, 2020. https://ieeexplore.ieee.org/document/9187209/

[20] P. K. Mvula, P. Branco, G.-V. Jourdan, and H. L. Viktor, "COVID-19 malicious domain names classification," *Expert systems with applications*, Oct. 01, 2022. https://www.sciencedirect.com/science/article/pii/S0957417422008715