Malicious URLs identification using Bagging and Boosting Algorithm

Student Number: 20069009 Student Name: Doma chaitanya Goud

Course: MSc Computer Science (Modular) with Advanced Research

Supervised by: (Chidinma Chiejina)

Contents

[1 Introduction 2](#_Toc147792263)

[1.1 Overview of Malicious URL and its adverse effects 2](#_Toc147792264)

[1.2 Bagging and Boosting Algorithm and its applications. 3](#_Toc147792265)

[1.3 Identifying Malicious URLs using Various Approaches 3](#_Toc147792266)

[1.4 Aim 4](#_Toc147792267)

[1.5 Research Question 4](#_Toc147792268)

[1.6 Objectives 4](#_Toc147792269)

[1.7 Problem Statement 5](#_Toc147792270)

[1.8 Current Issues 5](#_Toc147792271)

[1.9 Technical Feasibility 6](#_Toc147792272)

[2 Background research 8](#_Toc147792273)

[2.1 Introduction 8](#_Toc147792274)

[2.2 Need for Malicious URL identification 8](#_Toc147792275)

[2.3 Challenges in detecting Malicious URL 9](#_Toc147792276)

[2.4 Traditional methods used for detecting malicious URLs 9](#_Toc147792277)

[2.5 Overview of Bagging and boosting algorithms 10](#_Toc147792278)

[2.6 Application of bagging and boosting algorithms 10](#_Toc147792279)

[2.7 Recent research on malicious URL detection 11](#_Toc147792280)

[2.8 Linkage to aim 17](#_Toc147792281)

[3 Methodology 18](#_Toc147792282)

[3.1 Choice of methods 18](#_Toc147792283)

[3.2 Justification and Support of Choices 18](#_Toc147792284)

[3.3 Dataset Description 19](#_Toc147792285)

[3.4 Project Design 20](#_Toc147792286)

[3.5 Vectorization Techniques 20](#_Toc147792287)

[3.6 Random Forest 21](#_Toc147792288)

[3.7 Bagging Classifier 22](#_Toc147792289)

[3.8 AdaBoost 23](#_Toc147792290)

[3.9 Gradient Boost 24](#_Toc147792291)

[3.10 XGBoost 25](#_Toc147792292)

[3.11 Catboost 25](#_Toc147792293)

[3.12 Validation 26](#_Toc147792294)

[3.13 Consideration of Ethical/Legal/Professional and Social issues 28](#_Toc147792295)

[3.14 Considering Commercial and Economic Context 28](#_Toc147792296)

# Introduction

There is a growing danger to people's and businesses' online safety and privacy due to the proliferation of bad Uniform Resource Locators (URLs). Although machine learning techniques have been presented as a solution, researchers in this field face considerable hurdles due to the large number and variety of URLs, as well as the ongoing modification of malicious URLs. These URLs frequently mimic those of real sites, making them difficult to spot. The sophistication of malicious URLs has grown along with the development of new technologies. They may also employ social engineering techniques, such as sending an email that looks to have come from a reliable source, to deceive their targets into visiting the malicious website. These rogue URLs provide a risk of identity theft, financial loss, and brand harm. By using the same training data but assigning different weights to each instance of the basic model (usually decision trees), boosting is an ensemble approach (Mandela, Shaker, & Etyang, 2023) for improving predictive accuracy. When misclassification is minimized, machine learning models such as bagging and boosting perform better. Top features for every prediction made by the ML model may be identified with the aid of Shapley values, which accurately reflect the amount of contribution made by each feature (Kumar and Subbiah, 2022).

## Overview of Malicious URL and its adverse effects

A malicious URL is an address on the web that was created with the goal of causing harm. Cybercriminals and other bad actors create these URLs to trick users into downloading malware or otherwise harming themselves or their businesses. Visiting a malicious website can have serious consequences, such as the theft of sensitive information or the infection of a machine with malware. The use of phishing URLs is one of the most common forms of phishing assault and represents a serious risk to internet security. Finding these URLs is difficult since attackers are always changing their methods (Bu and Cho, 2023). The goal of creating professional social networks is to help their users find better jobs. Depending on its purpose, a website's forum might be open to the public or restricted to members of a specialised profession or with a shared interest. To target a specific individual, an attacker just needs the user's email address, which is readily available on social networking sites. Similar to emails offering a cash award, but with a malicious link, these messages may be spam (Jain,Sahoo and Kaubiyal, 2021).

## Bagging and Boosting Algorithm and its applications.

The two most popular families of integrated, or so-called ensemble, decision tree classifiers are bagging (an abbreviation for Bootstrap AGGregatING) and boosting, both of which rely on modifying training samples. With bagging, numerous predictors are developed separately and then integrated using model averaging techniques like weighted average or majority vote. Boosting is a method of EL in which the models are constructed in a sequential fashion, with each subsequent predictor being used to correct the mistakes produced by the preceding predictors (Jafarzadeh et al, 2021). In the fields of cybersecurity and fraud detection, bagging may be used to spot unusual patterns. Text classification activities like spam detection and sentiment analysis can benefit from the use of bagging. Although gradient boosting is a powerful technique that frequently produces high predictivity, its past uses in forecasting huge compound libraries or constructing in silico predictive models that require regular retraining have been constrained by the comparatively long computing time required to do so. An updated version of the gradient boosting technique called LightGBM (Zhang et al., 2019) takes advantage of a leaf-wise tree growth strategy to address the algorithm's scalability and computing time issues without sacrificing its excellent predictivity. After the XGB model has been trained, it and the aforementioned design constraints are fed into the DE method, where the parameters that may be adjusted to locate the global minimum or maximum of the considered objective function are defined (Lee et al, 2023).

## Identifying Malicious URLs using Various Approaches

A number of preventative measures such as receptive blacklisting of URLs, malicious URLs continue to provide a considerable threat. This is the case despite the development of solutions for their prevention. It is necessary to have a system of real-time detection of malicious URLs that is able to efficiently identify and prevent assaults of this nature. Phishing URLs typically feature certain relationships between the registered domain level and the path of the URL (Mehndiratta et al, 2023). A goal prediction value may be modelled using linear regression, and it is based on the independent variables. It fulfils the responsibility of forecasting the value of a variable called y, which is backed by an independent variable called x. Feature engineering is typically relied on by systems that search for harmful URLs using algorithmic approaches based on rules or on machine learning. This calls for knowledge and experience of the relevant field. Even after extracting characteristics from a dataset, it is possible that it may not fully capitalise on the potential of the dataset (Raja et al., 2023). There are three distinct vectorization approaches that are utilised in order to vectorize the URL text. Two distinct datasets (D1 and D2) that are often employed in the research domain were utilised in order to conduct an analysis of the effectiveness of the strategy that was suggested (As et al, 2023).

## Aim

This research focuses on developing a web security system that is based on bagging and boosting algorithms and has the ability to automatically recognize harmful URLs. It does this by making use of the lexical, network-based, and content-based features that have been extracted using feature extraction techniques. Also, the research focus on the effectiveness of the URL features when compared to vectorization techniques in detecting malicious URL detection.

## Research Question

* How effective are the URL features such as Lexical, DNS Statistical and Third-Party features help in detecting malicious URLs when compared to URL vectorization methods like TF-IDF, Count and Hashing vectorizers?
* Does boosting algorithms perform better than bagging algorithms such as in identifying URL text analysis?
* Is there any variations in the accuracy of the bagging and boosting algorithms when they are hypertuned?

## Objectives

* To collect URL dataset contacting Lexical, DNS Statistical and Third-Party features and to convert URLs to vectors by vectorization techniques such as TF-IDF, Count and hashing vectorizers.
* To implement a bagging and boosting algorithms such as random forest, bagging classifier, adaboost, gradient boost, XGBoost and catboost to identify malicious URLs with the use of extracted features and vectorized data.
* To enhance the performance of bagging and boosting algorithms in a web security system by hypertuning to ensure higher accuracy in malicious URL detection and prevention with at least 95% of accuracy.
* To compare the performance between the URL features techniques and vectorization technique by analyzing the performance result of bagging and boosting algorithms.

## Problem Statement

The goal of this research endeavor is to create a web security system capable of accurately identifying malicious URLs via the integration of bagging and boosting algorithms. The research assesses the relative performance of URL features like lexical, DNS statistical, and third-party features versus URL vectorization techniques in identifying malicious URLs. The project's specific objectives call for assembling a dataset consisting of URLs and employing feature extraction tactics like lexical analysis, DNS statistical analysis, and third-party analysis. In addition to other methods, TF-IDF, Count, and Hashing will be implemented to convert URLs into numerical vectors. Advanced machine learning techniques like bagging algorithms (random forest and bagging classifier), boosting algorithms (Adaboost, gradient boost, XGBoost, and catboost), will be utilized to detect harmful URLs by analyzing the extracted features and vectorized data. Hyperparameter fine-tuning can significantly enhance the predictive abilities of bagging and boosting algorithms in regard to malicious URL identification. Lastly, we'll assess the two methodologies in light of bagging and boosting algorithm results. Through this study, we hope to make a meaningful contribution to web security by creating a robust system for detecting and stopping malicious URLs, thereby safeguarding online users.

## Current Issues

“The digital world is becoming increasingly interconnected and cyberattacks such as phishing are becoming more common.” (Tareen, S.et al, 2022)

“The detection of malicious websites has become a critical issue in cybersecurity.” (Hu, Z.et al, 2023)

“Malicious URL, a.k.a. malicious website, is a common and serious threat to cybersecurity. Malicious URLs host unsolicited content (spam, phishing, drive-by downloads, etc.) and lure unsuspecting users to become victims of scams (monetary loss, theft of private information, and malware installation), and cause losses of billions of dollars every year. It is imperative to detect and act on such threats in a timely manner.” (Sahoo et al, 2017)

“Malicious URLs host unsolicited content and are used to perpetrate cybercrimes. It is imperative to detect them in a timely manner.” (Le et al, 2018)

The key objective of this investigation is to cultivate a cutting-edge web security system employing bagging and boosting methods for the automatic detection of harmful URLs. Of paramount importance are the lexical, network-based, and content-based features used to detect malicious URLs. In addition, the analysis assesses the performance of URL features relative to URL vectorization methods including TF-IDF, Count, and Hashing vectorizers for identifying harmful URLs. Examining the performance gaps between bagging and boosting algorithms in text analysis, particularly in relation to URLs, the study also considers the impact of hyperparameter tuning on their accuracy.

The intricate web of digital connections brought about by recent advancements in technology has sparked increased concerns regarding cybersecurity, as evidenced by the rising frequency of cyberattacks such as phishing (Tareen, S. et al., 2022). Cybersecurity now requires urgent attention to identifying malevolent websites (Hu, Z. et al., 2023). These obstacles highlight the necessity for cutting-edge web security measures that identify and hinder malevolent URLs with superior efficacy. This research project seeks to tackle these concerns through the integration of bagging and boosting algorithms, with a focus on evaluating the efficacy of various URL feature extraction techniques and vectorization methods in enhancing web security and mitigating potential risks posed by malicious URLs.

## Technical Feasibility

Machine Learning Algorithms: Cybersecurity experts frequently employ bagging algorithms (e.g., Random Forest, Bagging Classifier) alongside boosting algorithms (e.g., Adaboost, Gradient Boosting, XGBoost, CatBoost) to identify malicious URLs.

Algorithm Comparison: Researchers frequently comparatively evaluate feature-based and vectorization approaches for malicious URL detection in the cybersecurity field.

Data Collection: Obtaining URL datasets comprising relevant features and malicious URLs is indeed possible, thanks to accessible resources.

Algorithm Implementation: Data scientists are well equipped to tackle malicious URL detection with the aid of software libraries like scikit-learn.

# Background research

## Introduction

URLs that are known to lead to malicious websites can be detected through a method known as malicious URL detection. Malicious websites are ones that intend to cause harm to visitors by means such as data theft or virus installation. For identifying fraudulent URLs, efficient algorithms including bagging and boosting are utilized. They excel in efficiency, accuracy, and robustness against overfitting compared to other machine learning techniques. Some drawbacks are their intricacy and openness to interpretation (Le, et al, 2018). The below literature review discusses the need for malicious URL identification, challenges in detecting malicious URL, traditional methods used for detecting malicious URLs, overview of Bagging and boosting algorithms and recent researches carried out on malicious URL detection.

## Need for Malicious URL identification

Internet resources are identified using Uniform Resource Locators (URLs), which follow a standard syntax and structure. Attackers will often alter the structure of a URL in order to trick people into sharing a malicious link. These URLs lead to harmful resources or pages where hackers can install malware on victims' machines, steal personal information, or redirect visitors to other phishing sites. In addition, downloading links with malicious URLs can rapidly disseminate via email and instant messaging on public networks (Do Xuan, et al, 2020).

Cybersecurity threats from rogue URLs, often known as malicious websites, are widespread and pervasive. Users lose billions of dollars annually due to scams that involve losing money, having personal information stolen, and installing malware because of malicious URLs hosting spam, phishing, drive-by attacks, etc. Email, ads, online searches, and links to other websites are just some of the many possible sources of traffic to these sites. Timely detection and response to such risks is essential (Sahoo, et al, 2017). As of 2020, 80.7% of systems would have been infiltrated at least once, according to reports on the defence against cyber threats (Madhubala et al., 2022). Criminals using the epidemic as an excuse to launch a widespread attack via social media and email with links to malicious websites. To prevent these kinds of attacks, a good defence system and a solution that can properly categorize and identify dangerous URLs is needed (Janet and Kumar, 2021).

## Challenges in detecting Malicious URL

Attackers use novel methods, such as phishing, to deceive targets into revealing private data by visiting phony websites. Due to the semantic-based anonymity of the attack architecture, the sheer number and variety of URLs, and the vulnerabilities they exploit, detecting rogue sites is a difficult task. While several solutions have been presented to counter phishing, the ongoing emergence of new assaults remains a significant barrier to effective malicious URL detection (da Silva Monteiro and Sampaio Rocha, 2023). Even the most vigilant Internet users may have trouble identifying phishing URLs. As millions of new websites are being created every day, all of which require users to create login profiles before they can access protected content, it is becoming increasingly difficult to determine which of these sites can be trusted. Many phishing websites are launched every 20 seconds (Potpelwar, 2023). This challenging problem needs an effective solution like blacklist, rule-based detection, anomaly-based detection, etc. to identify malicious websites (Venkatesh, 2023).

## Traditional methods used for detecting malicious URLs

The blacklist technique, the heuristic method, and the anomaly-based method are the most common approaches to identifying dangerous URLs. The following is an explanation of these.

Blacklist method: A common method is the blacklist method, which screens incoming URLs against a list of potentially malicious URLs. It is a quick and straightforward process. The reason it is still utilized as the standard approach to find these URLs is because it is easy to implement. This strategy is ineffective for new malicious sites that are constantly being created. Almost 92% of infected websites are not flagged or blacklisted by the common search engines (Velpula, et al, 2022).

Heuristic approach: Phishing website detection using a heuristic technique relies on a set of established rules. It's a more sophisticated use of the blacklist strategy. In order to detect unknown phishing URLs in real time, rule-based detection can be simple and proactive. Data mining algorithms and other forms of detection training are also used in certain methods. Despite its potential, it may not be as effective against emerging or sophisticated phishing attempts unless updated often (Chinguwo and Dhanalakshmi, 2023). Feature extraction and feature representation are only two examples of the machine learning stages that are implemented to efficiently detect malicious URLs and help overcome this constraint.

Anomaly-based detection: Phishing websites may be detected with the use of an anomaly-based detection method, which compares their features to those of known, trustworthy websites. It takes a two-pronged strategy by first analyzing users' eye movements while they view a web page with mobile eye tracking, and then using an anomaly-based detection algorithm to spot phishing URLs (Kushwaha et al., 2017).

Traditional methods of managing security are showing limitations due to the increasing number of new security risks, the rapid development of new information technology, and the severe shortage of security people. To overcome these demerits, Bagging and boosting algorithms are implemented for effective detection of malicious URLs.

## Overview of Bagging and boosting algorithms

According to research conducted by Odegua (2019), ensemble approaches are among the most often used ways for enhancing the predictive capacity of a ML model. An ensemble is made up of a collection of independently educated base learners or models, the predictions of which are blended when applying the ensemble to new examples. The generation of ensemble models may often be broken down into two distinct paradigms: sequential ensemble techniques and parallel ensemble methods. The sequential ensemble technique involves the generation of the base learners in a sequential fashion, with Ada-Boost serving as a popular representation. The fundamental goal behind the parallel ensemble approach, of which Bagging is a sample example, is to take advantage of the independent predictive capacity of the individual base learners. This is due to the fact that the error may be minimized by combining the abilities of these individual base learners. Multiple learning algorithms, or base learners, are taught on a collection of training samples in a process known as "bagging." The term "bootstrap aggregation" refers to this type of sampling. Depending on the data collection, the predictive power increase brought about by any ensemble may at times be small or may even go in the opposite direction. It is recommended to use Boosting rather than Bagging for most ensembles. Boosting may experience overfitting when there is noise present, which may explain part of the performance decline that Boosting has seen.

## Application of bagging and boosting algorithms

In the research carried out by Taser et al. (2021), methods of ML and ensemble learning are used in order to forecast illnesses. Diabetes mellitus is a long-term condition that results in an abnormally elevated level of sugar in the blood. The author has applied bagging and boosting methods to experimental data and used six distinct DTB classifiers (C4.5, random tree, REPTree, decision stump, Hoeffding tree, and NBTree) in order to make an early prediction of diabetes. When the individual DTB classifiers' prediction results were analyzed, it was discovered that the NBTree method had the greatest accuracy score of all the classifiers, coming in at 96.74%. In addition, the findings suggest that the bagging and boosting strategies perform better when employing an NBTree as a base learner and produce the highest accuracy rate of 98.65%.

## Recent research on malicious URL detection

An artificial intelligence model was utilized by the authors of the paper (Tiryaki et al., 2023), with the goals of working with a sizable dataset and achieving the best possible results in the detection of fraudulent URL addresses. A seven-layer RNN model was utilized, and two comparable national and international datasets were joined and merged to produce a large new dataset that consisted of 579,112 URL addresses. This new dataset was then employed. After that, the newly created data set is segmented into a training set and a test set. The first data set was used for training, and the second data set was used as a test once it had been processed. The processing of this data set ended up yielding a success percentage that was greater than 91%. The detection rate at this rate is quite high, which is a very good result for finding fraudulent URL addresses. It would be increasingly significant if effective methods for recognizing hazardous sites were adopted as internet usage increased so that users could be protected from targeted cyber assaults. This would allow people to feel more secure when browsing the internet.

A diagram of different types of url

Description automatically generated

Figure 1 Accuracy table of (Tiryaki et al., 2023)

The goal of this study (Deekshitha et al., 2022) is to use a variety of classification and boosting methods, such as bagging, Gradient boosting, and cat boosting, to improve precision. Website features from the UC Irvine ML Repository are used to extract the features. ML methods have been imported using the Scikit-Learn tool. In a ratio of 80:20, the dataset is split into training and testing sets. Classifiers are learned with the help of a training set, and then their efficacy is measured with the help of a testing set. Accuracy scores, false negative rates, and false positive rates have been calculated to assess classifier performance. Compared to the decision tree and support vector machine methods, the results demonstrate that the detection accuracy provided by the cat boost classifier is higher (97.2%) and that provided by the gradient boost classifier (97.4%), while the false negative rate is lower. When using 90% of the data as a training dataset, all classifiers perform well. When more information is used as training data, classifiers perform better. The framework developed in this study is thought to be useful for the development of automated systems capable of accurate categorization of phishing websites.

A white grid with black numbers

Description automatically generated

Figure 2 Results of (Deekshitha et al., 2022)

The classification of URLs was accomplished with the assistance of three well-known ensemble algorithms (Mandela et al., 2023). They are referred to as Random Forest, Bagging, XGBoost, and AdaBoost, respectively. In order to pre-process them, a dataset containing labelled URLs is utilized. Accuracy and total run time are the metrics that are utilized in the process of determining how well the models work. According to the findings, each of the four algorithms is capable of producing satisfactory results, with accuracy ratings that are higher than 95%. Random Forest had the greatest accuracy, followed by XGBoos, Bagging, and AdaBoost in that order. The quickest method was XGBoost, which had a runtime of 1 minute and 36 seconds. As a result, it is an option that is acceptable for use in real-time applications. The authors give intuition into the relative capabilities of various ensemble algorithms on the task of URL categorization and show the need to pick a suitable model dependent on the individual features of the data and the prerequisites of the application. However, the properties of the data as well as the needs of the application should guide the selection of the appropriate method.

(Naim et al., 2023) developed a robust and sustainable detection strategy that is based on the extraction, analysis, and learning of design aspects with the purpose of identifying harmful websites. In order to identify fraudulent websites, a new generation of algorithms with an expanded design and attribute learning has been developed. This algorithm is able to learn about and assess the structures, contents, and looks of web pages, as well as their reputations. The Mendeley Data Repository contains the dataset that was created and analyzed and is available for users to use. The results of a large-scale experiment that was carried out on more than 35,000 websites reveal that the suggested algorithm efficiently detects more than 83% of all dangerous websites while retaining a low false-positive rate of 2%. These results were obtained from the experiment that was carried out. In addition, the model that was developed takes into account input from users and notifies administrators of newly discovered malicious websites; as a result, it is capable of successfully fending off zero-day assaults.

Developing a content-aware malicious website detection (CAMD) algorithm is the objective of this project (Chang et al., 2023). This CAMD technique uses an innovative contextual visualization procedure to determine if a website is dangerous. This method uses convolutional neural networks to identify dangerous websites by first retrieving their essential codes and then transforming them into one-dimensional grayscale visuals. In order to determine whether or not the suggested CAMD would be practical, the website VirusTotal was combed through for both regular and dangerous URLs. Based on the findings, it was determined that the CAMD that was proposed obtained an accuracy of 98%.

Research conducted by Abdul Samad et al. (2023) uses data-parity, hyper-parameter optimization, and feature selection are the three types of tuning parameters. In the experiment, researchers employ two different datasets culled from online resources including the Mendeley repository and the UCI repository in addition to the eight most popular machine learning algorithms. The findings indicate that data balancing contributes only a minimally increased level of accuracy, but adjustments to hyperparameters and feature selection make a considerable contribution. When all of the parameters that need to be fine-tuned are combined, the performance of ML algorithms may be increased, allowing them to exceed previous research efforts. The findings demonstrate that adjusting parameters makes machine learning systems more effective. Both Random Forest (RF) and Gradient Boosting (XGB) are able to obtain accuracy rates of 97.44% and 97.47% for Dataset-1, respectively. In the case of Dataset-2, the accuracy values attained using Gradient Boosting (GB) and Extreme Gradient Boosting (XGB) are 98.27% and 98.21%, respectively.

A graph of a graph with text

Description automatically generated with medium confidence

Figure 3 Comparison of accuracy in dataset 1 of (Abdul Samad, et al, 2023)

A graph of different types of data

Description automatically generated

Figure 4 Comparison of accuracy in dataset 2 of (Abdul Samad, et al, 2023)

Table 1 Existing work on malicious URL detection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Author | Research area | Dataset used | Algorithm used | Accuracy |
| Tiryaki, et al, 2023 | Detection of fraudulent URL addresses | Dataset of 579,112 URL addresses | Seven-layer RNN model | 91% |
| Deekshitha, 2022 | Phishing Website Detection | UC Irvine Machine Learning Repository | Gradient Boosting,Cat boosting | XGB-97.4%  Cat boosting-97.2% |
| Mandela, et al, 2023 | URL categorization | Dataset containing labeled URL | RF,Bagging,XGBoost,AdaBoost | Random Forest-95% |
| Naim, et al, 2023 | Identification of harmful websites | Mendeley Data repository |  | 83% |
| Abdul Samad, et al, 2023 | Malicious URL detection | Mendeley and UCI repository | Fine-tuned ML (RF, XGB) | Dataset 1  Rf-97.44%  XGB-97.47% |
| Dataset 2  RF-98.27%  XGB-98.21% |
| Chang, et al, 2023 | Malicious websites detection | VirusTotal | CAMD | 98% |

## Linkage to aim

The above literature review indicates the diversity of methods for identifying fraudulent URLs. All the way from straightforward rule-based procedures to sophisticated machine learning algorithms including bagging and boosting, are analysed. Moreover, different feature selection methods are utilized by different researchers. This makes it difficult to determine which features are most important for malicious URL detection.  In the context of this, feature extraction involves identifying and extracting features from URLs that are predictive of maliciousness. However, these feature extraction techniques are not yet explored in malicious URL identification. Also, in harmful URL detection, URL vectorization techniques could be employed to convert raw URL data into a format that is more amenable to machine learning algorithms by representing each URL as a vector of features, where each feature is used to describe a unique property of the URL. These vectorization techniques and performance comparison between feature extraction and vectorization techniques are also not yet explored for malicious URL detection.

# Methodology

## Choice of methods

The following algorithms and techniques are used for the detection of malicious URLs using the bagging and boosting algorithms, as explained below.

Textual data can be converted into numerical vectors using a method called TF-IDF. It does this by assigning weights to terms according to the frequency with which they appear in a document in comparison to a corpus of documents. The importance of terms contained in URLs can be determined using TF-IDF vectors (Jiang et al., 2021). AdaBoost is an ensemble method that works to improve classification accuracy by compensating for the shortcomings of the base classifiers. It does this by giving instances that have been incorrectly classified a higher weight, which enables subsequent base classifiers to pay more attention to these instances (Ibrahim et al., 2023). These are more advanced boosting algorithms, and they optimise gradient boosting in order to improve both the efficiency and accuracy of the process. They are equipped with regularisation methods and have the ability to effectively deal with missing data (Bentéjac et al., 2021). The general ensemble technique known as "bagging" involves generating multiple subsets of the dataset and then training base classifiers on each of those subsets. In most cases, the ultimate prediction is arrived at by adding up the results of all of these different base classifiers (Tüysüzolu et al., 2020). In the Domain Name System (DNS), statistical features involve the analysis of data associated with URLs that is related to DNS. This may include information such as the total number of DNS requests, the popularity of the domain, or patterns of DNS querying (Khormali et al., 2021).

## Justification and Support of Choices

“With the D2 dataset, DT with TF-IDF vectorizer obtains a greater accuracy of 99.5 %.” (Raja et al, 2023)

“Besides, in the case of IDSs, Gentle and Real AdaBoost show the same performance as they have about 70% lower error rates compared to Modest Adaboost, however, Modest AdaBoost is about 7% faster than them.” (Shahraki et al.,2020)

“Experimental results revealed that Adaboost with SVM has outperformed best among the classification methods by achieving the highest accuracy 97.61%.” (Subasi and Kremic, 2020)

“We also contrast our En\_Bag model with state-of-the-art solutions and show its superiority in binary classification and multi-classification with accuracy rates of 99.3% and 97.92%, respectively.” (Abu and Al-Fayoumi, 2023)

“The experimental outcome confirmed that the RF and GB classifiers are better choices for the said problem. Since, majority of the malicious activity detected by the developed model, it can be said that the ML-based algorithms are a better option for the prevention of DNS attacks on DoH traffic.” (Singh and Roy, 2020)

Various bagging and boosting algorithms, including Decision Trees, AdaBoost versions, Support Vector Machines, and ensemble models, have been shown to be useful in this research for the purpose of identifying harmful URLs. Each individual algorithm or ensemble method has its own advantages and can be fine-tuned for a variety of applications in cyber security.

## Dataset Description

Dataset1:

The dataset chosen for this research can be downloaded using the below link.

<https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset>

This large dataset consists of 651,191 URLs, of which 428103 are considered benign, 96457 are considered defacements, 94111 are considered phishing, and 32520 are considered malware. This dataset is a compilation of data from several sources. The data is collected from a URL dataset (ISCX-URL-2016) that has samples of benign, phishing, malware, and defacement URLs. The Malware domain black list dataset s used to increase phishing and malware URLs. The number of safe URLs is increased using faizan git repo and the quantity of phishing URLs is increased utilising the Phishtank dataset and the PhishStorm dataset. In order to consolidate only the URLs and their class type, the collected URLs from several sources are merged. The final dataset contains two columns: a URL in one and a type representing the type of maliciousness in another.

Dataset 2:

Second dataset contains URL features i.e., Lexical, DNS Statistical and Third Party features

<https://www.unb.ca/cic/datasets/dns-2021.html>

This is a large DNS features dataset of 400,000 benign and 13,011 malicious samples processed from a million benign and 51,453 known-malicious domains from publicly available datasets. The malicious samples span between three categories of spam, phishing, and malware. This dataset, namely CIC-Bell-DNS2021 replicates the real-world scenarios with frequent benign traffic and diverse malicious domain types.

Three set of features are extracted here. Lexical features include 12 features such as subdomain, Top-level domain, Second-level domain etc. DNS based features contains country details, ASN and TTL values etc. Third party features include domain name, registrar name, domain age etc. This dataset contains Malware, Spam, Phishing and Benign URLs.

## Project Design

A diagram of a software algorithm

Description automatically generated

Figure 5 Project design

## Vectorization Techniques

A machine is only able to comprehend numerical data, hence it is impossible for it to operate with categorical data. Therefore, in order to provide a computer with a text corpus, categorical data must be translated into numerical value. The term "vectorization" refers to this technique. TF-IDF, Count and Hashing vectorizers are explained as below:

TF-IDF: In the field of natural language processing (NLP), the approach known as Term Frequency-Inverse Document Frequency is the one that is utilised the most frequently for transforming text documents into a matrix representation of vectors. The predominance of a term in a collection of documents relative to an individual document is reflected in the tf-idf representation of those documents. It is possible to build effective search engines on the basis of the capacity of tf-idf scores to reflect the prominence of terms in the text and to capture the relevance of the document to a particular search query. The computation of the Inverse Document Frequency (IDF) score, on the other hand, is vocabulary-dependent; this limits the usefulness of tf-idf scores to corpora that are subject to ongoing change (Singh and Shashi, 2019).

Count Vectorizer: The Count Vectorizer provides an easy technique to tokenize a variety of content archives, create a jargon of known terms, and encode fresh reports utilising that jargon. Additionally, the Count Vectorizer provides a means to generate a vocabulary of known words. An encoded vector is a return that contains the length of the whole vocabulary as well as a whole integer that represents the total number of times that each word appeared in the record (Smitha and Bharath, 2020).

Hashing vectorizer: Hashing Vectorization is a process that transforms a document containing a collection of texts into a sparse matrix that holds token occurrence counts. In order to locate the token stream name and feature integer index mapping, this text vectorizer method makes use of a hashing technique (Haque, Manik and Hashem, 2019).

## Random Forest

Random Forest is a computation based on tree-based learning for the ensemble. Constructs numerous decision trees and then combines them in order to provide forecasts that are both more accurate and stable. Bagging is the training approach that was used. Through the use of row sampling and feature sampling, the high variance that was produced from the decision tree was turned into a low variance. With the help of hyperparameter, determining the number of decision trees is possible (Wang et al, 2019).

A diagram of a tree

Description automatically generated

Figure 6 Structure of Random Forest (Zhang et al,2023)

It is what's known as an ensemble algorithm, and what it does is aggregate several calculations of the same or similar sort for the purpose of classifying things. RF provides the benefits of being able to analyse high-dimensional data, having excellent anti-noise abilities, and avoiding the problem of overfitting. These are the advantages of using RF (Sagi and Rokach, 2018).

## Bagging Classifier

The term "bagging" refers to a framework for the algorithm that trains a number of distinct models, one after the other, and then lets all of the models vote to test the output of samples. Bagging makes use of a sampling method called replacement sampling in order to produce several training subsets, which are then put to use in the process of training classifiers, as is seen in Figure. Because each step of the training process may be treated as a standalone activity, the overall procedure might be sped up using parallel computing. Specifically, the training subset of Bagging is picked at random, which means that various subsets have a chance of having the same data in them (Sun et al, 2018).

A diagram of a classifier

Description automatically generated

Figure 7 Structure of Bagging classifier ((Zhang et al,2023)

In addition to this, the training of each classifier is made more unpredictable by the use of bagging. Following training, all classifiers are integrated in order to limit the amount of variation in the outcomes of the prediction. Bagging is able to effectively enhance the discrimination accuracy, particularly when there is a substantial amount of variation between the variables. This is because the expectation of numerous weak classifiers is better than that of the strong classifier (Ngo, Beard and Chandra, 2022).

## AdaBoost

AdaBoost will choose, within the training set, the important classification feature set for a significant number of times. The component weak classifier is trained one step at a time, and then the system chooses the weak classifier with the most accurate threshold. In the last step of the process, a powerful classifier is built by selecting the weak classifier that performs the best in each iteration. On the other hand, AdaBoost builds a robust classifier by combining many fewer effective classifiers. There is an unequal difference in the weights of each of the weak classifiers, and the higher weight will be given to the classifier that is stronger. Because it is difficult for it to manage noisy data and outliers, AdaBoost requires a high-quality dataset (Soui et al, 2021).

A diagram of a data processing process

Description automatically generated

Figure 8 AdaBoost (Salcedo-Sanz et al, 2022)

In the field of machine learning, the AdaBoost algorithm has developed into a key feature classification strategy. It is now widely utilised in a broad range of applications, some of which include face recognition and image retrieval, intrusion detection, object recognition, and feature extraction, amongst others (Ding et al, 2022).

## Gradient Boost

The machine language technique of gradient boosting is an effective tool for resolving problems involving data that is diverse, noisy, and complicated. It does this by employing binary decision trees as its primary predictors, and it possesses the robust qualities of minimising the likelihood of data overfitting and reducing the need for hyperparameter adjustment. It does this by combining a gradient boosting decision tree, often known as a GBDT, with categorical features, concentrating on categorical variables, and resolving issues of gradient bias and prediction shift (Ogar, Hussain and Gamage, 2022).

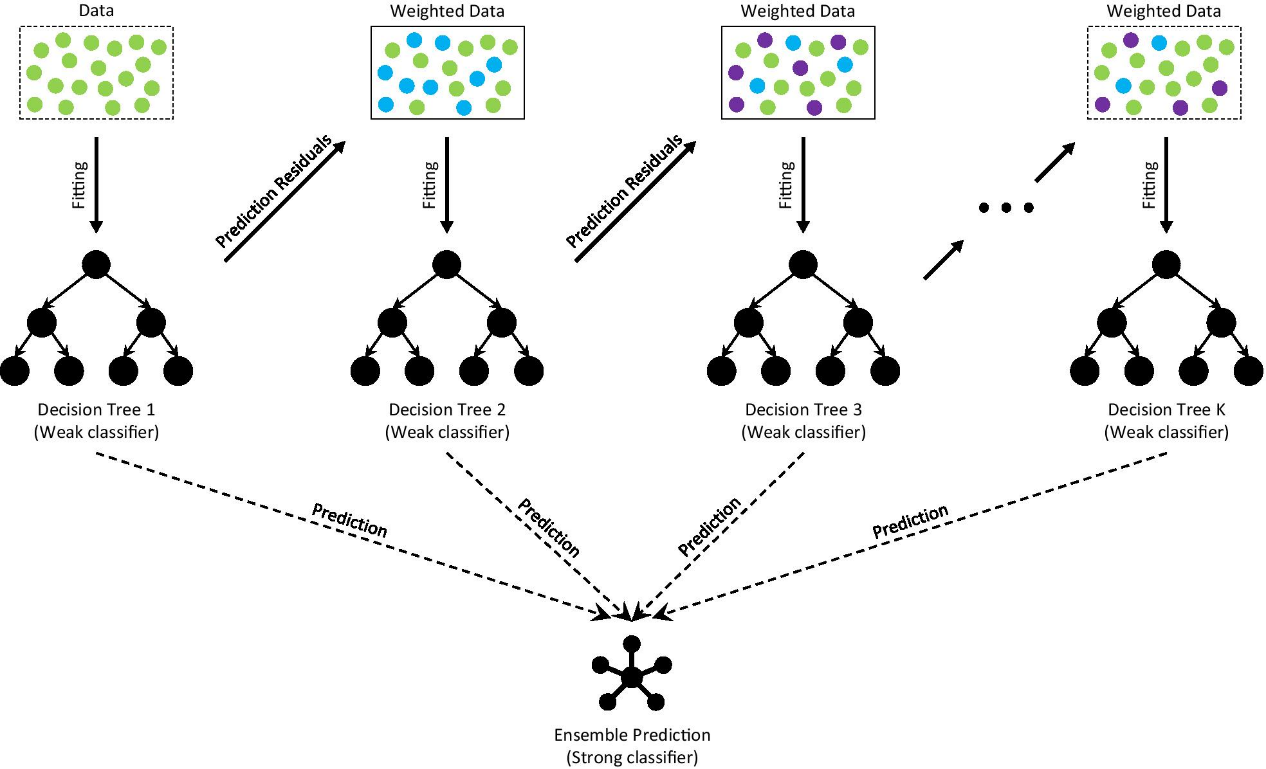


Figure 9 The architecture of Gradient Boosting (Deng et al, 2021)

It is possible to improve the algorithm's stability by training it using all of the sample datasets that are currently accessible. After determining the goal value of the model, each sample's characteristics were then modified, and after that, the sample's weight and priority were added (Deng et al, 2021).

## XGBoost

XGBoost has replaced GBM as the superior data preparation tool because of how well it constructs predictive models for both regression and classification. XGBoost is a machine learning algorithm that has been widely praised for its superior performance on a wide variety of datasets. The major benefit of XGBoost is its ability to solve restrictions by combining models, which speeds up the learning process (Asselman, Khaldi, & Aammou, 2021).

A diagram of a tree

Description automatically generated

Figure 10 XGBoost (Wang, Chakraborty and Chakraborty, 2020)

An extremely powerful Gradient increase. Developed for processing massive and complex datasets. Regularised boosting is an ensemble strategy that avoids overfitting. It may be scaled to any situation. Its ability to process sparse input and to participate in parallel and distributed computation accelerates the learning process (Hajihosseinlou et al, 2023).

## Catboost

CatBoost is an example of a GBM-based algorithm that stands out in a notable way in the literature due to its distinctive properties. It does this by presenting an innovative ordered boosting strategy as a solution to the problem of prediction shift, which results in noticeable and robust support for categorical features. CatBoost places an emphasis on in-tree splitting while also providing a variety of options for the management of categorical characteristics. When there are just a few categories available for a feature, CatBoost employs a method known as one-hot encoding (Sergio Gonzalez et al., 2020).

A diagram of a data processing process

Description automatically generated

Figure 11 CatBoost (Niaz Muhammad Shahani et al, 2022)

It is also possible for it to translate categorical qualities into numerical ones, and this is helped along by the frequency of category occurrences. It does this by exchanging the categories for the average goal values, which is a strategy that is more complex. In addition to enhanced learning phases and a balanced symmetry, these trees have a decreased susceptibility to overfitting. CatBoost excels at managing missing values, and this further strengthens the platform's resilience as well as its usefulness in a variety of settings (Sujoy Barua et al, 2021).

## Validation

Confusion Matrix: Effortless tuning leads to optimized classifications suitable for diverse contexts through efficient parameter modification. This tool offers crucial information about misclassified instances, directing subsequent revisions for increased precision and output.

A diagram of negative negatives

Description automatically generated with medium confidence

Figure 12 Confusion Matrix (Jeppesen et al, 2019)

Precision: Invaluable in accurately distinguishing between safe versus harmful websites, precision serves within information retrieval frameworks. Evaluated by its capacity to exclude perilous URLs without producing unnecessary alerts.

Recall: Mandatory is revisiting relevant knowledge regarding URL catching; specifically, recollection's importance lies within precision-focused enhancement process aimed towards prudent decision production. Employing this mechanism, damaging websites are accurately pinpointed while misses are kept to a bare minimum.

F1 Score: Combining aspects of precision and recollection, the F1 score cultivates a comprehensive evaluation technique striving toward a proficient union between precision and simplicity, the sytem formidably tests URL detection adeptness.

Accuracy: The accuracy with which the system labels URLs reflects its broader success in safeguarding internet users. The system's overall correctness receives thorough examination.

A math equations on a white background

Description automatically generated

Figure 13 Evaluation Metrics (Punn and Agarwal, 2020)

Training Time: Resources must somewhat compromise to maximize model precision's theoretical output; this concerns training duration. Practicality depends on well- questioned training; thus, enhanced security setup evolution is assured.

Testing Time: Timely considerations for testing come with an inherent balance dilemma where quick judgments meet profound appraisals. Allowing for rapid defense against harmful web addresses, this function reinforces internet safety.

## Consideration of Ethical/Legal/Professional and Social issues

The primary objective of this project is to investigate whether or not the use of feature extraction techniques is more effective than the application of natural language processing to URLs in the task of identifying malicious URLs of a domain. However, there are ethical concerns regarding the data set used in this project as well as the algorithms that were used in it. The data sets that were utilized in this project were obtained from the websites of UNB and Kaggle respectively. On the basis of the data set that was collected, the experiments will be carried out; however, any manipulation of the data in any way could result in inaccurate output. Altering the architecture of the algorithms that are being used in this project is not recommended because doing so could result in inaccurate output. Users or patterns that lead to the identification of individuals will not be identifiable after the implementation of algorithms. The most significant social repercussions of this project, primarily with regard to the URL that was utilized in this research. Because of safety concerns, the URL that was used in this investigation will not be tested on any browser nor will it be executed to any browser or shared. This is a common project that focuses primarily on applying algorithms in order to construct an effective system for determining which URLs contain malicious content. This project is not going to be used for any one particular group of people or business in particular. The findings of this research will assist law enforcement agencies and network administrators in improving their respective applications. However, the findings of the research are subjected to testing in the community. Therefore, it is recommended that this deduction system be tested before it is implemented in a real-world setting. When using the algorithms, it is important to be aware of their limitations and to apply them in a responsible and ethical manner.

## Considering Commercial and Economic Context

Cybersecurity Solutions: A robust web security system supported by sophisticated algorithms can be marketed as a cybersecurity solution. Organizations can use this system to protect their networks and users from malicious URLs.

E-commerce Security: As cybersecurity is a worldwide problem, it presents opportunities for global partnerships and research collaborations, with e-commerce platforms serving as a united front against online threats.