



# An intelligent identification and classification system for malicious uniform resource locators (URLs)

Qasem Abu Al-Haija<sup>1</sup> · Mustafa Al-Fayoumi<sup>1</sup>

Received: 23 December 2022 / Accepted: 5 April 2023 / Published online: 20 April 2023  
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2023

## Abstract

Uniform Resource Locator (URL) is a unique identifier composed of protocol and domain name used to locate and retrieve a resource on the Internet. Like any Internet service, URLs (also called websites) are vulnerable to compromise by attackers to develop Malicious URLs that can exploit/devastate the user's information and resources. Malicious URLs are usually designed with the intention of promoting cyber-attacks such as spam, phishing, malware, and defacement. These websites usually require action on the user's side and can reach users across emails, text messages, pop-ups, or devious advertisements. They have a potential impact that can reach, in some cases, to compromise the machine or network of the user, especially those arriving by email. Therefore, developing systems to detect malicious URLs is of great interest nowadays. This paper proposes a high-performance machine learning-based detection system to identify Malicious URLs. The proposed system provides two layers of detection. Firstly, we identify the URLs as either benign or malware using a binary classifier. Secondly, we classify the URL classes based on their feature into five classes: benign, spam, phishing, malware, and defacement. Specifically, we report on four ensemble learning approaches, viz. the ensemble of bagging trees (En\_Bag) approach, the ensemble of k-nearest neighbor (En\_kNN) approach, and the ensemble of boosted decision trees (En\_Bos) approach, and the ensemble of subspace discriminator (En\_Dsc) approach. The developed approaches have been evaluated on an inclusive and contemporary dataset for uniform resource locators (ISCX-URL2016). ISCX-URL2016 provides a lightweight dataset for detecting and categorizing malicious URLs according to their attack type and lexical analysis. Conventional machine learning evaluation measurements are used to evaluate the detection accuracy, precision, recall, F Score, and detection time. Our experiential assessment indicates that the ensemble of bagging trees (En\_Bag) approach provides better performance rates than other ensemble methods. Alternatively, the ensemble of the k-nearest neighbor (En\_kNN) approach provides the highest inference speed. 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.

**Keywords** Machine Learning · Uniform resource locators (URLs) · Benign URLs · Malware URLs · Spam URLs · Phishing URLs · Defacement URLs · Detection · Classification

## 1 Introduction

The spread of mobile devices, IoT technologies, ad hoc networks, and intelligent sensors has resulted in a surge in the usage of the global Internet due to the rapid development of global networking and communication

technologies. Due to the COVID-19 pandemic, which has relocated many daily activities such as social networking, e-banking, e-commerce, and cyberspace, the year 2020 has seen a tremendous shift in internet usage. According to worldwide data, roughly 5.25 billion internet users access and use the internet often, with internet use rising by 1.355 percent for the global population from 2000 to 2022, which is now projected at 7.9 billion people [1].

As a result of the exponential growth in the Internet and digital technology usage, web apps have become pervasive in many commercial sectors and a vital part of people's

✉ Qasem Abu Al-Haija  
q.abualhaija@psut.edu.jo

<sup>1</sup> Department of Cybersecurity, Princess Sumaya University for Technology (PSUT), Amman, Jordan

everyday lives and activities worldwide. While the Internet infrastructure is an open, anonymous, and unsupervised design that allows for a suitable system for cyber assaults, it simultaneously exposes network systems and computer users to many security and significant risks.

In today's environment, rogue URLs are a pervasive threat to many internet businesses. Even though the World Wide Web has become one of the most essential on the Internet, it has dramatically increased the likelihood of cyber assaults. The hostile actors have used the Internet to carry out harmful operations such as phishing, spamming, and malware. For example, phishing involves sending an email that looks to have come from a trustworthy source to trick people into clicking a URL (Uniform Resource Locator) the email and leads to a bogus webpage. Moreover, a malicious URL is created to promote frauds, attacks, and scams; by clicking on an infected URL, a malware can be downloaded on your device, or one can be tricked into providing sensitive and private information on a bogus website. However, many of these programs are either vulnerable to web defacement attacks or designed and controlled by hackers. Scam websites and phishing schemes are two examples. It is critical to detect websites containing harmful code to limit the spread of malware and defend end users from becoming victims.

In practice, the most known attacks using malicious URLs are spam and phishing. Phishing attacks are a form of cybercrime in which the intruder attempts to get private and critical information from the end user, such as login credentials or account information through email or fake websites that look legal. For instance, Figure 1 shows a phishing website identical to Twitter, so the attacker can manipulate the user to enter their credentials.

The most current research of the APWG Phishing Activity Trends Report indicated a steady flow of phishing

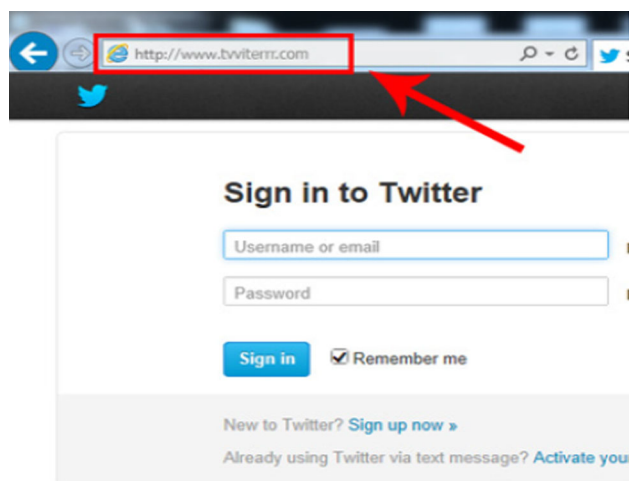


Fig. 1 Phishing website example

and confirmed attack sites in the first three months of 2022 [2]. The number of malicious URLs and their trend are displayed in Fig. 2, which covers the period from the 1st quarter of 2019 through the 1st quarter of 2022. Website builders and free web hosting companies have seen a rise in phishing attacks. Total phishing assaults for the first quarter of 2022 were 1,025,968, according to the company. Additionally, the APWG has the worst phishing quarter ever recorded for the first three months of 2022. In March 2022, the APWG experienced a monthly assault total of 384,291 for the first time.

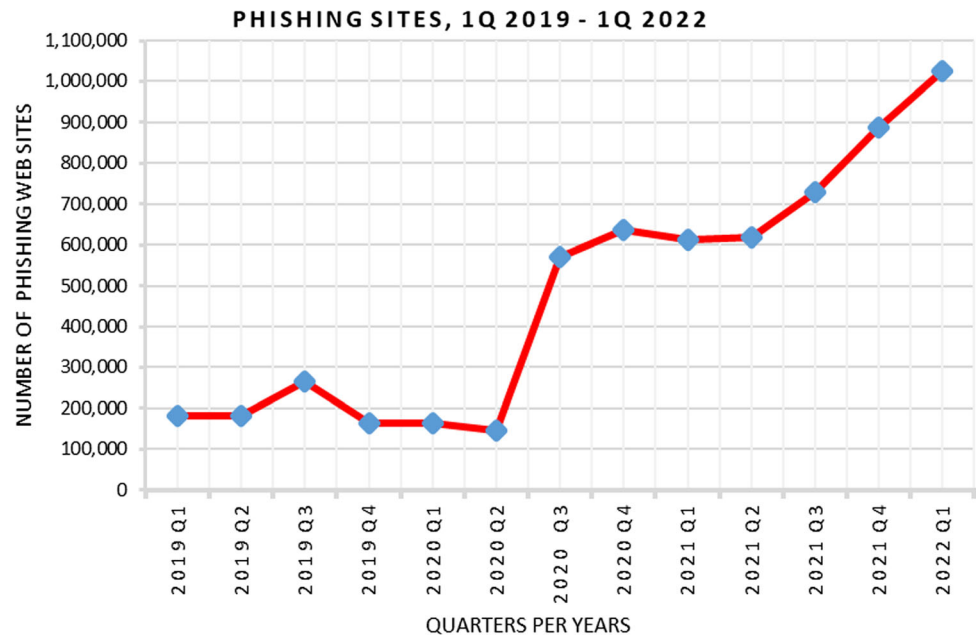
URLs are usually divided into two types: those that are safe and those that are bad. Malicious URLs are also put into different groups based on the type of attack, like spam, phishing, or malware. Detecting malicious URLs has been a major issue for cybersecurity professionals for many decades. Several strategies for detecting malicious URLs and protecting end users from becoming victims of an attack are offered [3–8]. These solutions are classified according to their detection type, which can be feature-based detection or blacklist-based detection. Features representing URLs are automatically extracted and examined in feature-based detection, whereas blacklist-based detection depends on user reports and expert analysis.

Malicious URL detection is critical because several attackers distribute malicious links to genuine websites, for instance social media platforms and emails. Furthermore, certain harmful URLs are distributed via downloading malware, which is able to infect the detector while it is crawling. Furthermore, because of the striking similarities between certain malicious web content with genuine information, such as phishing websites and fraudulent websites, malicious URL detection is more efficient and accurate than web content detection.

URL-based detection is preferred. It is a proactive and secure strategy for the detecting machines because it can detect malicious URLs prior to the user visiting them. This makes URL-based detection the preferred method.

The classification of malicious URLs to identify attack types is significant since it might indicate the direction and necessity of network security measures. For instance, defaced URLs may require a faster response than spam URLs due to credential breaches. In addition, spamming may be easier to overlook; malware infestation requires quick action. Identifying and classifying URLs that lead to dangerous websites is a significant problem with real-world implications. It is possible for a business to protect itself with a machine learning model by blocking incoming emails and employee websites based on harmful URLs. In addition, the identification and classification of malicious URLs are further effective for real-time threat detection and applications with limited resources, such as mobile and Internet of Things (IoT) devices.

**Fig. 2** Phishing URLs number reported to APAC from the first Quarter of 2019 to the first Quarter of 2022



Consequently, in this paper, an intelligent Identification and Classification System for Malicious URLs is proposed. One of the main goals of this system is to detect malicious URLs and identify attack types of malicious URLs attempt to launch including benign phishing, spam, malware, and defacement. Therefore, this system performs two tasks: The first challenge involves binary classification, whereas the second involves multi-label classification. The dataset includes information from five distinct URL categories, which are as follows:

- Benign: trustworthy websites that offer specific services and include no potentially hazardous data or information.
- Phishing: the act of a website pretending to be a reliable entity in an electronic connection to get information, personal information like usernames, passwords, and credit card numbers.
- Malware is defined as software created to interfere with, damage, steal confidential data, or gain unauthorized access to a computer system.
- Defacement is a form of online vandalism that occurs when an unauthorized person modifies the appearance of a website by using various ways.
- Spam is the dissemination of uninvited and irrelevant content for advertising, phishing, malware distribution, etc.

## 2 Literature review

Several studies are given on how to find malicious URLs and keep users from falling victim to an attack. This study examines URL-based detection solutions and summarizes

studies that examine malicious URLs. In addition, several patterns observed in the literature and research gaps will be explored.

Sayamber and Dixit [9] have proposed a model using an NB classifier to identify and classify four types of classes (Spam, Malware, phishing, and Benign). Clustering and classification techniques were applied to assess the NB classifier. In [10], the authors have used the Weka tool as the ML infrastructure, applying five different classification techniques (PRISM, Multi-Layer Perceptron (MLP), Naïve Bayes (NB), KStar (K\*), and Random Forest (RF)) against their dataset acquired from the NASA repository. Their classification process was divided into six main steps: Identifying the required data, feature selection, identifying the training set, classifier selection, training, and classifier evaluation using the test set. Two experiments were directed using tenfold cross-validation; one using features selection and the other one without it. In [11], the authors have used an intelligent system for detecting phishing attacks using some data mining techniques to classify websites as authentic or fraudulent. Several ML classifiers were used to classify phishing websites in the dataset that was obtained from the UCI repository, these classification algorithms are Rotation Forest (RoF), K-Nearest Neighbor (kNN), C4.5 Decision Tree, Support Vector Machines (SVM), Artificial Neural Networks (ANN), and Random Forests (RF).

In [12], the authors proposed an approach that uses natural language processing (NLP) techniques to detect phishing attacks by performing semantic analysis of the text to detect malicious content of phishing emails. SEA-Hound, the name of the proposed system, examines a one

sentence at a moment of a document, and returns “True” if the document includes a social engineering attack. A topic blacklist which is a list of (verb-direct object) pairs, was generated using ML by developing a MultinomialNB() function which produces a prediction label for each (verb-direct object) pair and generates a confidence score (0,1) for the prediction. For training they have used 2000 emails (1000 phishing emails and 1000 non-phishing), and 10014 emails (5014 phishing emails and 5000 non-phishing) for testing purposes.

In [13], the authors discuss three approaches for detecting phishing URLs and websites using ML by; analyzing several features of URL (black or white list), checking the who is hosting and managing the website (heuristics), and analyzing the visual appearance of the website (Visual Similarity). They have developed a system that takes advantages of all the three approaches by; Monitor all “http” traffic, comparing the domain of every URL with the white/black list, analyzing the whole website features, checking the similarity between the phishing page and the target page, extracting and comparing the CSS of suspicious and legit pages, using ML classification ML algorithms like logistic regression, decision tree, and random forest against the collected data, conducting similarity score and checking it against a pre-defined threshold. D. Patil and J. Patil [14] reported that none of the writers who have worked in this field have employed multi-class classification to determine the nature of URLs, instead focusing solely on binary classification. They claim that the URLs are not classified as any other sort of spam, just as malicious or benign. In order to identify the nature of URLs, the basic concept more precisely behind their work was to develop a multi-class classification method for detecting harmful attacks based on confidence weighted learning (a multi-class classifier) and retrieved 42 new phishing, malware, and spam features from malicious URLs using supervised machine learning strategy for training.

Adebowale et al. [15], in their research have used integrated features of images, texts, and frames in building their model for web-phishing detection and prevention scheme. Adaptive Neuro-Fuzzy Inference System (ANFIS) based robust scheme was deployed. Also, Combine features images, text, and frames for phishing detection. SVM was used for classification and detection. Ping Yi et al. [16] developed a framework for detecting phishing websites using Deep Learning (DL). Two main features-types were intended for phishing websites: original features and interaction features. Deep Belief Networks (DBN) is the ground of the proposed detection model.

In [17], the authors have used the ML stacking model in their proposed framework in order to detect phishing websites. They started by using features reduction

algorithms (Relief-F, Information Gain [IG], Recursive Feature Elimination [RFE], and Gain Ratio [GR]) against the dataset to end-up with two sub-sets, strongest features and weakest features. Then normalize the data and feed it into the Principal Component Analysis (PCA). Afterward, ML algorithms (Naïve Bayes, random forest [RF], bagging, neural network [NN], support vector machine [SVM], and k-nearest neighbor [KNN]) were applied against the two sub-sets. Later on, staking occurs by merging the best performing algorithms, two stacking models were used, one using RF+Bagging+NN, and the other one using RF+Bagging+KNN. The authors choose the tenfold cross-validation as the training model, then using the performance evaluation measures to assess the detection rate.

In [18], Sahingoz et al. projected a real-time anti-phishing system which employs seven distinct classification algorithms and natural language processing (NLP)-based characteristics. Their system is claimed to execute real-time detection, new phishing websites detection, can distinguish a large number of legit/phishing websites, third-party independent, and language independent. They have used KNN, SMO, Decision Tree, Kstar, Random Forest, Adaboost, and Naive Bayes as ML algorithms, and run each one in three different classes using NLP Features, Word Vector, and Hybrid. Rao et al. [19] used the ensemble ML to classify benign and malicious URLs using the XGBoost (gradient boosted decision trees implementation) algorithm. Hot encoder was used to change categorical values to values that machine understand, which enhance the building and predicting speed. Rohan in his research [20], has used distributed modern machine learning technologies using Spark MLlib. This model uses logistic regression and Support Vector Machine (SVM) techniques to estimate the trustworthiness of a URL.

In [21], the authors presented a phishing detecting data-driven framework by using deep learning multilayer Perceptron approach. The framework is consisted of several steps, starting by data acquisition, then data preprocessing for selecting the most effective features and analyzing them using IG, RFE, GR, and Relief-F, afterward the data is normalized, then PCA is applied. Now the classification phase is conducted by training the model using a tenfold cross-validation technique along with the Multilayer Perceptron (MLP) model.

In [22], the authors proposed a model for phishing detection using an intelligent detection model by employing Multilayer Perceptron (MLP), and 70% for training and 30% for testing. They begin with data acquisition, data cleaning, and normalization. After that comes the processing phase, in which ranking the features according to their significance, then using the MATLAB Independent significance Features test (IndFeat()) for ranking purposes and using the Correlation-based Feature Selection

(CfsSubsetEval()) to evaluate the attributes. To utilize the best features for classification, an intersection between output features from IndFeat() and CfsSubsetEval() is performed. Finally, applying MLP classifier to learn the correlation between the features and apply this approach against the testing dataset.

In [23], the authors used ML and binary visualization to build their approach by employing an automated detection process. In the learning phase, ML algorithm TensorFlow and samples are built, while in the detection phase, the submitted URLs are tested against the samples in the database. To distinguish between legit and phishing websites, the model visualizes the scraped HTML file to a 2D images (using BinVis tool) which are processed by the TensorFlow model in which analyzes the images against the training model. The system stores any URL that pass-through in a database to be visualized in the background. If a user visits a URL, the system will check if it does exist in the database, if not it will scrape the HTML code and store it in the database, then visualizing the HTML file into a 2D image, and finally analyzes it to perform the detection.

Xiao et al. [24] considered using CNN-LSTM along with multi-head self-attention (MHSA) in their proposed model called “CNN-MHSA” intended for detecting phishing websites. The first step is using CNN to learn the URL features automatically and considering the relationship of characters inner dependencies in the URLs. Then MHSA explores these relationships to assign weights for CNN studied features. In [25], Shirazi et al. employed Adversarial Autoencoder (AAE) in their proposed model to augment existing datasets used by machine learning algorithms. AAE mimics phishing websites by generating samples and providing a system of measurement to measure the quality of those samples. The samples were evaluated against models trained using real-world data and were able to escape the detection model. Some of these samples were then used in training. The main contribution is AAE and synthesized data in the training set.

In [26], the authors present system for detection phishing websites by analyzing the URL for patterns. ML techniques were employed to learn URL’s data patterns. Phishing Websites Dataset (PWS) was used, consists of small balanced dataset and a large, unbalanced dataset. The phases for this system are preprocessing (data collection, feature selection, conversion of data, encoding, data normalization, shuffling, augmentation and distribution), training (ML algorithm for pattern characterizing. 70% training and 30% testing), and detection (model deployment and prediction). Wide Shallow Neural Network (W-SNN), Narrow Shallow Neural Network (N-SNN), and the Optimizable Decision Tree/GentleBoost Ensemble (ODT) were used for learning and optimization module, and Sigmoid classifier for detection module.

In [27], Maini et al. has built a model to classify legit and phishing websites using several ML classifiers; Ada-Boost, Decision tree, KNN, Logistic Regression, Random Forest, Extreme Gradient Boosting (XGBoost), Support Vector Machines (SVM), and Naive Bayes. An ensemble model was built aiming to improve the detection rate. Wang et al. [28] proposed a model using Dynamic Convolutional Neural Network (DCNN) for malicious URLs detection. Stating that their model can prevent attackers from passing the current detection model, and the problem of not having the malicious features extracted properly is solved. They have added a new folding layer replacing the pooling layer of the original multilayer convolution network with a k-max pooling layer. This layer is adjusted dynamically depending on the URL length and the convolution layer depth. A new embedding method is proposed to learn the vector representation by leveraging the word embedding which is based on the character embedding. Several experiments were conducting (Character embedding alone, Word embedding alone, and Word embedding based on character embedding) proving that their model is outperformed the overall results. In [29], Manoj et al. has proposed an intelligent ML model to detect phishing websites by assessing the characteristics of the phishing site and selecting an acceptable combination of systems for classifier training. Logistic regression, SVM, XGBoost, and Auto-encoders were used as classifiers.

In [30], the authors rely on ML for email phishing classification. PCA was used as an attribute selector, Ranker Search as a search method, and Multilayer perceptron (MLP), DT, and Logistic Model Tree (LMT) as classification algorithms. They have used the PCA and Ranker to govern the positively classified, incorrectly categorized, and Kappa statistics for each algorithm, then apply the classification algorithms as pairs (PCA + MLP), (PCA + J48), and (PCA + LMT) against the resulted data. To sum up, Table 1 provides a comparative summary of all the surveyed studies.

By examining past research and solutions that analyze URLs for harmful content, and the patterns observed in the literature, it is highlighted that most existing research has only concentrated on malicious URL detection. Although the detection process is a huge challenge, classifying malicious URLs is crucial as it can indicate the direction and importance of measures an organization should take to secure its network. There is widespread agreement that blacklisting malicious URLs is ineffective because it does not address issues like obfuscation or new attacks. Consequently, the majority of current research has been on applications of machine learning techniques. Among the most popular algorithms, Random Forest (RF) seems to be one of the most effective, Decision Tree, X-Boot, and so on.



**Table 1** Summary of review-related research

Ref	Model	Datasets	Classes	Advantages	Limitation
Sayamber et al. [9]	NB	DMOZ, Yahoo!, jwSpamSpy, and DNS-BH	4-Classes	Fast 4-classes were classified Good features extraction	Not effective real-time implementation Only two classifiers were used Shortened and TORs URLs might not be detected
Ibrahim et al. [10]	RFC_FS	Phishing Websites Features 2015	2-Classes	Fast, Deep Learning was used for better results	Small Dataset Shortened and TOR's URLs might not be detected
Subasi et al. [11]	RFC	UCI Phishing Websites 2015	2-Classes	ANN and Rotation Forest provide high accuracy	Small Dataset Shortened and TOR's URLs might not be
Peng et al. [12]	MNB	Joseph phishing 2014	2-Classes	Several ML classifiers were used Semantic text analysis is a fast approach NLP was used for the detection	Heavy-weight due to Nural Network detected Small Dataset, Shortened and TOR's URLs might not be
Patil et al. [13]	RFC	Alexa.com, rank2traffic.com, and siterankdata.com	2-Classes	Covering many phishing features Analyzing the visual appearance of the website Monitors all traffic on the end-user system	Many legit pages are blockage (many false positives) Modern new phishing techniques might bypass the system
D. Patil [9] [14]	DT and majority voting technique	Alexa Top sites, Malware Domain List, jwSpamSpy	2-Classes and 4-Classes	Reliable and effective Several classifiers were used Multi-class classification	More discriminative spam URL attributes must be investigated Disguised Java Scripts on Webpages are not analyzed Dark web URLs cannot be classified
Adebowale et. al [15]	ANFIS + SIFT	UCI Phishing Websites Data Set (2015)	3-Classes (legit, suspicious, phishing)	Reliable Developing their ANFIS algorithm Client-side (browser extension)	Not so fast if implemented in a real-time environment ANFIS's computing cost is considerable due mainly to its complicated structure and gradient learning
Ping Yi et al. [16]	DBN	Real IP flows from ISP	2-Classes	DBN is a powerful deep learning technique Real-time implemented Good features extraction and analyzing	DBN has hardware specifications DBN demands a large amount of data to execute better procedures DBN is costly to train due to its sophisticated data models The system is heavy-weight
Zamir et al. [17]	KNN-RF-Bag	Phishing website dataset (Kaggle) 2017	2-Classes	Many classification and feature-reduction algorithms were used Deploying stacking for better results	Heavy-weight system Small Dataset Real-time implementation is not efficient

**Table 1** (continued)

Ref	Model	Datasets	Classes	Advantages	Limitation
Sahingoz et al. [18]	DT, KNN AdaBoost, K*, RF, NLP SMO, NB,	Private Ebbu2017 Phishing Dataset (73,575 URLs)	2-Classes	NLP, word vector, and hybrid-based features Language independent Several ML classifiers New Websites detection ability	The results can be enhanced using deep learning Shortened and TOR URLs can bypass
Rao et al. [19]	XGBoost, One hot encoder	Kaggle repository of malicious and benign URLs	2-Classes	High accuracy XGBoost is a powerful classifier Using One hot encoding Website visualization	This approach is heavy-weight Cannot detect TORs URLs Time-consuming
Rohan [20]	Logistic regression, SVM	URL Reputation Data Set	2-Classes	New and reliable approach, Fast	If any node fails, the reliability is down Heavy-weight system Complex
Saha et al. [21]	MLP	Kaggle	3-Classes	Multilayer deep learning approach Real-time implementation is applicable High accuracy	Small dataset Heavy-weight model MLP is sensitive to feature scaling
Odeh et al. [22]	MLP	PhishTank archive, MillerSmiles archive, and Google searching operators	2-Classes	Multilayer deep learning approach Very high accuracy Good features extraction	Heavy approach Not very fast
Barlow et al. [23]	MB_CNN	Private (25 samples)	6-Classes (2-Classes)	High reliability if trained correctly, Multi-level AI and visualization were used	Resource consuming slow performance in real-time application Not covering language adaptation, Small dataset
Xiao et al. [24]	CNN-MHSA	Private	2-Classes	Automated model, Detecting many phishing websites	Time-consuming in real-time implementation The URL length parameter may affect the model's resilience High accuracy
Shirazi et al. [25]	(AAE) Synthesized Data	Four datasets	2-Classes	Auto-Encoders reduce the dimensionality of the data for easier classification, Tested against four different datasets	AAE is not so fast The whole model is slow
Al-Haija et al. [26]	En_ODT	OpenDNS (PhishTank) and (Amazon) Alexa	2-Classes	Fast URL classification Many classification algorithms (ML/DL) were used	Relying on URL patterns only Shortened and TOR's URLs might not be detected Heavy-weight
Maini et. al [27]	RD, DT, NB, AdaBoost, KNN, XGBoost, SVM, Ensemble	Private	2-Classes	XGBoost is a powerful algorithm Fast Approach, Applying ensemble model Good features extraction	The main focus was on feature extraction only It might get bypassed in modern ways

**Table 1** (continued)

Ref	Model	Datasets	Classes	Advantages	Limitation
Wang et. al [28]	DCNN	Private	2-Classes	Good learning mechanism There is a low possibility of mixing benign and dangerous URLs High accuracy	The model is heavy-weight Dark web URLs cannot be classified In real-time, the detection process will affect the network performance
Manoj et al. [29]	Logistic regression, SVM, XGBoost, MLP, and Auto-Encoders	Alexa, Dmoz, personal web history, PhishTank, whois	2-Classes	Several classifiers were used Using XGBoost and MLP Good analyzing process	Small dataset Low accuracy Slow in real-time Relying on 3rd-parties
Rana et al. [30]	LMT	Phish Tank (2015)	2-Classes	Multilayer deep learning Approach for better detection	Old and small dataset, Modern phishing URLs might not be detected

### 3 Proposed classification system

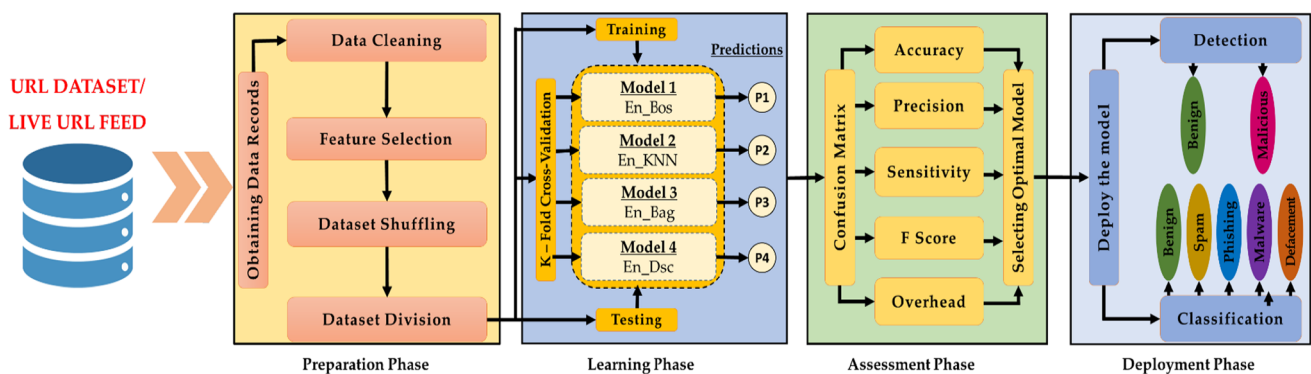
Identifying the malicious URLs and classifying them into attack types is a substantial task as it facilitates the proper network security measures and policies. Defaced URLs require a swifter response than spam ones since they infringe the credentials. Therefore, a number of intelligent and statistical techniques have been utilized to develop an autonomous and self-reliant detection system that can recognize and categorize malicious URLs to network administrators in security-based decision-making. In this work, we propose a detection and classification system that makes use of different ensemble learning approaches to develop a high-performant detection and classification for malicious URLs. The proposed system is illustrated in Fig. 3, and it is decomposed into four main processing phases: Preparation, Learning, Assessment, and Deployment. Besides, the dataset in this study, namely known as, URL dataset or ISCX-URL2016 [31], is composed of 57,000 URL samples distributed as Benign URLs (over 35,300), Spam URLs (over 12,000), Phishing URLs

(Around 10,000), Malware URLs (over 11,500), and Defacement URLs (over 45,450). Moreover, the dataset is formulated using 79 features and two-class labels (Binary and Multiple). This includes the query length, domain token count, the path token count, and others.

### 4 Preparation phase

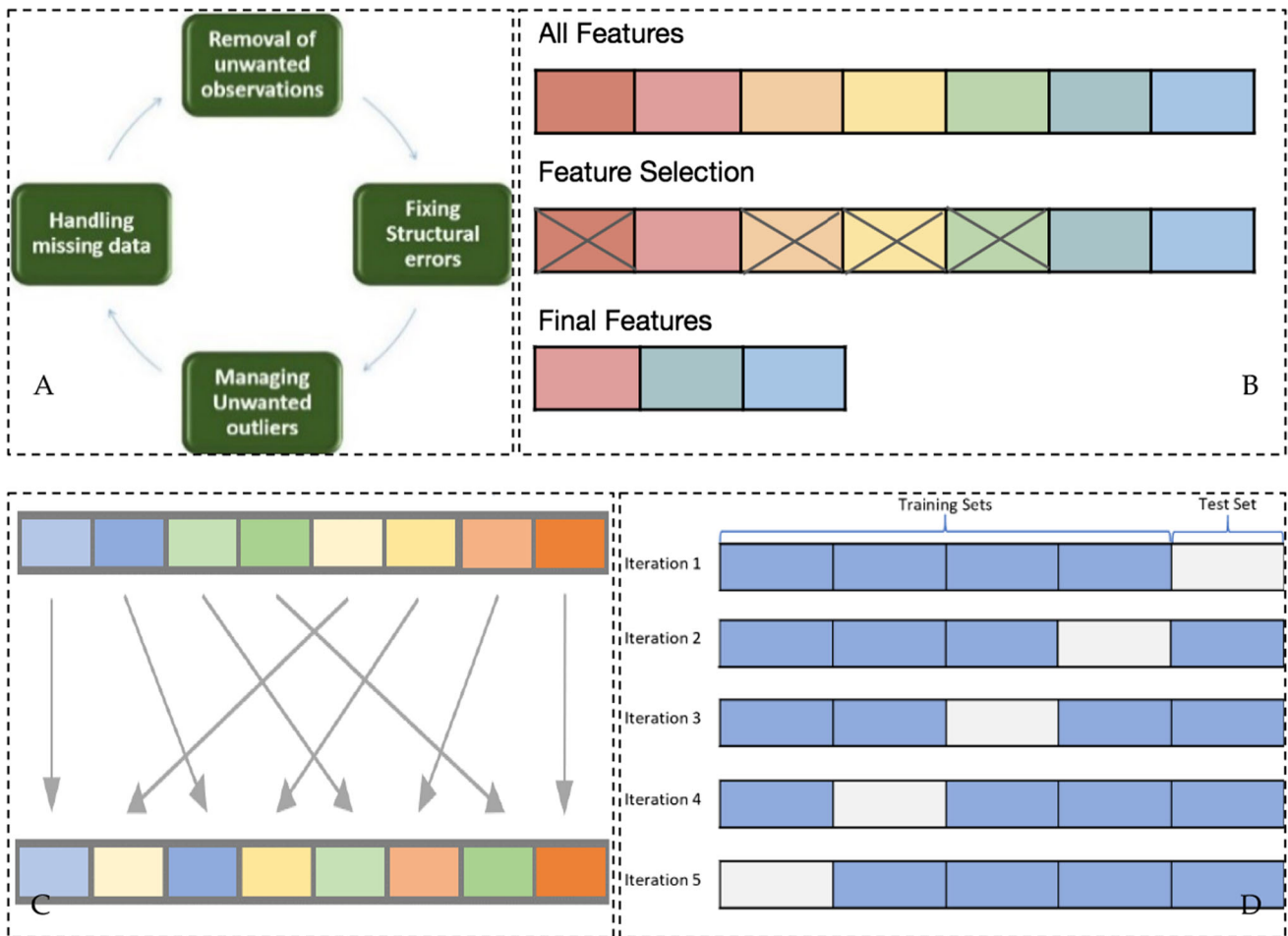
This stage is responsible for preparing the dataset for the learning process through the following operations:

- **Data Cleaning:** This process is responsible for providing an error-free dataset to be ready for feeding the learning phase. This phase includes removal of unwanted observation (duplicate/redundant or irrelevant values deletion), missing data handling (fixing issues of unknown missing values), fixing structural errors (fixing problems with mislabeled classes, types in names of features, the same attribute with different names, and others), and managing unwanted outliers



**Fig. 3** The dataflow diagram of the proposed Malicious URL Detection and Classification System





**Fig. 4** Data preparation phases (from top-left toward bottom-right): **A** Data cleaning, **B** Feature Selection, **C** Dataset Shuffling, and **D** Dataset Division

(unwanted values which are not fitting in the dataset). The data cleaning process is illustrated in Fig 4A [32].

- **Feature Selection:** This process is responsible for reducing the input features to the learning model by using only relevant features and eliminating irrelevant features. The process of feature selection is illustrated in Fig 4B. In this work, 59 features are selected out of 79 using the minimum redundancy and maximum relevance (MRMR) algorithm [33], where each feature is ranked based on minimum redundancy and maximum relevance and assigned an importance score [34].
- **Dataset Shuffling:** This process is responsible for mixing data and preserving logical relationships between columns. It randomly shuffles data from a dataset within a set of features (columns) [35]. The process of Dataset Shuffling is illustrated in Fig 4C.
- **Dataset Division:** This process is responsible for splitting the dataset into training datasets to train the learning model and testing the dataset to evaluate the learning model’s performance. In this research, we split

the dataset into 70% training and 30% for testing datasets. To ensure efficient validation, we have implemented fivefold cross-validation, which randomly split all data into five folds by dividing the data into different 70:30 ratios for the dataset at every fold. The model performance is evaluated at every fold. This procedure is repeated five times. After that, the overall performance metrics are calculated by averaging the results obtained from the five experiments (Folds/Iterations). The process of Dataset Division using fivefold cross-validation is illustrated in Fig 4D [36].

### 4.1 Learning phase

This phase is the main subsystem of the proposed malicious URL detection and classification system since it is accountable for creating the intelligent portions of the system. This can occur by training and validating the system using supervised learning methods that can then develop behavioral contours for the different types of

URLs (Benign, Spam, Phishing, Malware, and Defacement). In this work, ensemble learning techniques were employed to provide both detection and classification for URL websites. Two main reasons to use ensemble learning are (1) Performance (an ensemble can make better predictions and perform better than any single contributing model). (2) Robustness (an ensemble reduces the spread or dispersion of the predictions and model performance) [37, 38]. Specifically, four ensemble learning models were established and assessed to differentiate their performance and select the most performant model. These models are the ensemble of bagging trees (En\_Bag) approach, the ensemble of k-nearest neighbor (En\_kNN) approach, the ensemble of boosted decision trees (En\_Bos) approach, and the ensemble of subspace discriminator (En\_Dsc) approach. Table 2 outlines summarized points of the implemented models and their descriptions

## 4.2 Assessment phase

In this research, we have evaluated the performance of proposed models using a conventional machine learning assessment system of measurement. This includes the confusion matrix analysis (reporting TP, TN, FP, and FN), in which we calculate the number of correctly classified samples (NCS) and the number of misclassified samples (MCS) that can be used to derive the other standard metrics such as detection/classification accuracy, detection/classification precision, detection/classification recall (sensitivity), and the detection/classification harmonic average (F Score) [39]. Finally, we also measure every developed model's detection/classification overhead (prediction time). To sum up, Figure 5 summarizes the calculations for the performance assessment measurement system utilized in this paper.

## 4.3 Deployment phase

System deployment entails the transition of the developed service (such as the Malicious ULS detection system) to the eventual end-user and the transition of support and maintenance liabilities to the post-deployment support organization or organizations. The proposed Malicious ULS detection system is to be deployed with the intrusion detection/prevention system (IDS/IPS) as either host-based IDS, network-based, or even at both sides. Preliminary industry-standard IDS deployment imposes utilization of network-based IDS, then host-based IDS. This guarantees the network and host devices are protected [40]. The primary basis of any corporation is the network infrastructure, then devices within those networks. IDS should be deployed in the same fashion. Figure 6 illustrates the Deployment of IDS/IPS on an Enterprise deployment of IDS/IPS on an enterprise.

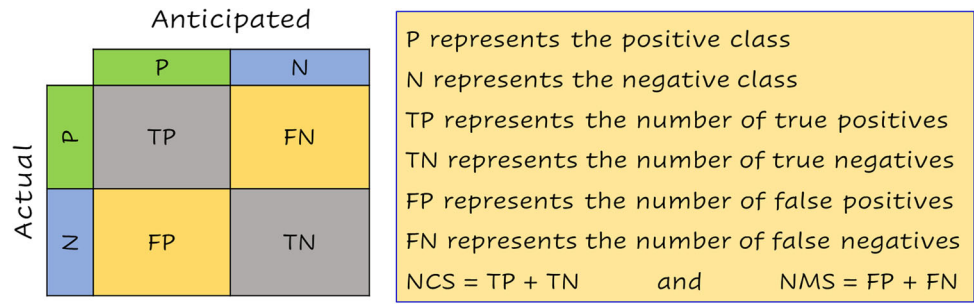
## 5 Results and analysis

Like any autonomous system, we evaluate our system using the above-mentioned conventional evaluation metrics. First, we provide the system evaluation using four ensemble learning models, viz. the ensemble of bagging trees (En\_Bag) approach, the ensemble of k-nearest neighbor (En\_kNN) approach, and the ensemble of boosted decision trees (En\_Bos) approach, and the ensemble of subspace discriminator (En\_Dsc) approach. Then, we extract the model that can be optimally used to provide detection and classification of the malicious URL. Thus, we further investigate more results to gain insights into the solution approach. Finally, we benchmark our best results with state-of-the-art models to gain more insights into the advantage of the proposed solution over existing models.

**Table 2** Summary of ensemble learning models utilized in this work

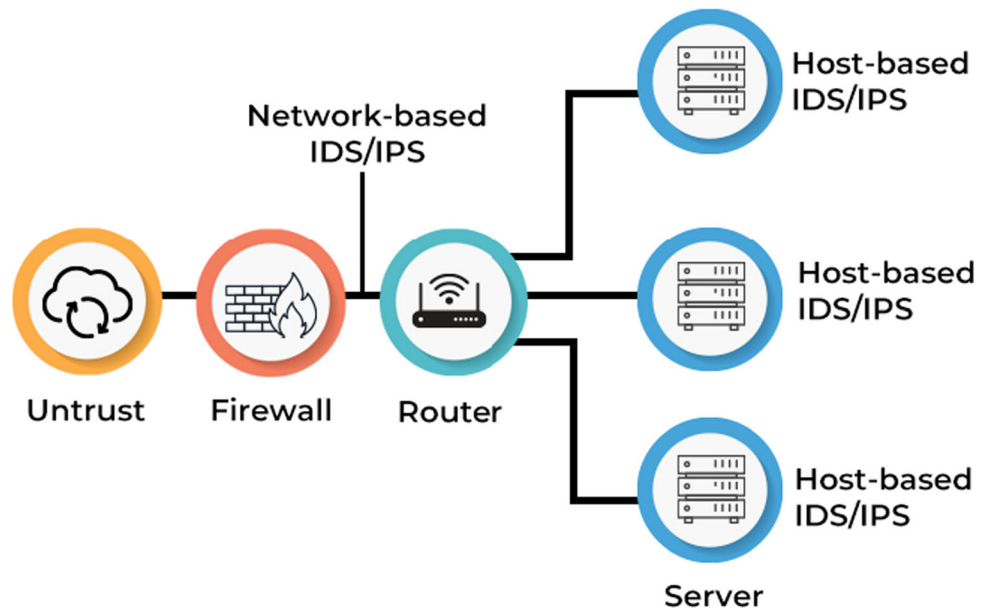
	En_Bag	En_Bos	En_kNN	En_Dsc
Ensemble method	Bag	AdaBoost	Subspace	Subspace
Learner type	Decision tree	Decision tree	Nearest neighbors	Discriminant
Max No. of splits	15,366 Splits	20 Splits	Hamming	Dimension 40
No. of learners	30	30	57 neighbors	30
Learning rate	0.001	0.1	0.1	0.1
Number of learners	10	10	10	10
Optimizer	Bayesian optimizer	Bayesian optimizer	Bayesian optimizer	Bayesian adam
Cost function	Minimum classification error	Minimum classification error	Minimum classification error	Minimum classification error

**Fig. 5** Summary of the performance assessment system of measurement



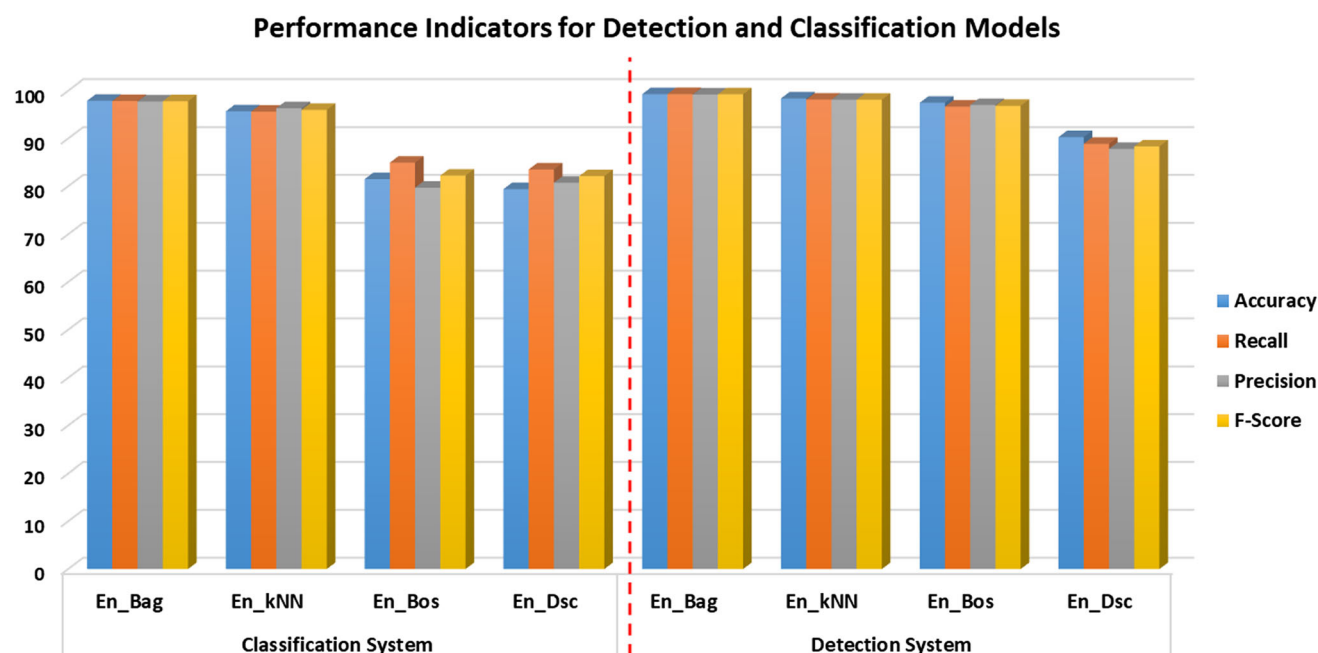
$\text{Accuracy} = \frac{\text{NCS}}{\text{NCS} + \text{NMS}} \times 100\%$	$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100\%$
$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\%$	$\text{F Score} = 2 \times \frac{\text{PPV} \times \text{TPR}}{\text{PPV} + \text{TPR}} \times 100\%$

**Fig. 6** The typical deployment of IDS/IPS on an enterprise



**Table 3** Summary of experimental results for the performance of using Model 1(En\_Bag), Model 2 (En\_kNN), Model 3 (En\_Bos), and Model 4 (En\_Dsc)

	Classification system				Detection system			
	En_Bag	En_kNN	En_Bos	En_Dsc	En_Bag	En_kNN	En_Bos	En_Dsc
Accuracy	97.92 (± 0.03)	95.70 (± 0.06)	81.50 (± 0.10)	79.40 (± 0.09)	99.30 (± 0.11)	98.40 (± 0.05)	97.50 (± 0.03)	90.30 (± 0.07)
Recall	97.87 (± 0.05)	95.64 (± 0.07)	84.95 (± 0.09)	83.52 (± 0.07)	99.32 (± 0.05)	98.17 (± 0.07)	96.69 (± 0.04)	88.89 (± 0.08)
Precision	97.77 (± 0.03)	96.36 (± 0.05)	79.71 (± 0.11)	80.74 (± 0.08)	99.23 (± 0.07)	98.14 (± 0.03)	97.03 (± 0.06)	87.82 (± 0.05)
F Score	97.82 (± 0.04)	96.00 (± 0.06)	82.24 (± 0.10)	82.12 (± 0.11)	99.27 (± 0.06)	98.15 (± 0.05)	96.86 (± 0.05)	88.35 (± 0.07)
D_Time	11.77 μSec	3.3 × 10 <sup>-3</sup> μSec	07.70 μSec	66.67 μSec	06.67 μSec	2.6 × 10 <sup>-3</sup> μSec	05.63 μSec	35.70 μSec



**Fig. 7** Summary of experimental results for the performance of using Model 1(En\_Bag), Model 2 (En\_kNN), Model 3 (En\_Bos), and Model 4 (En\_Dsc)

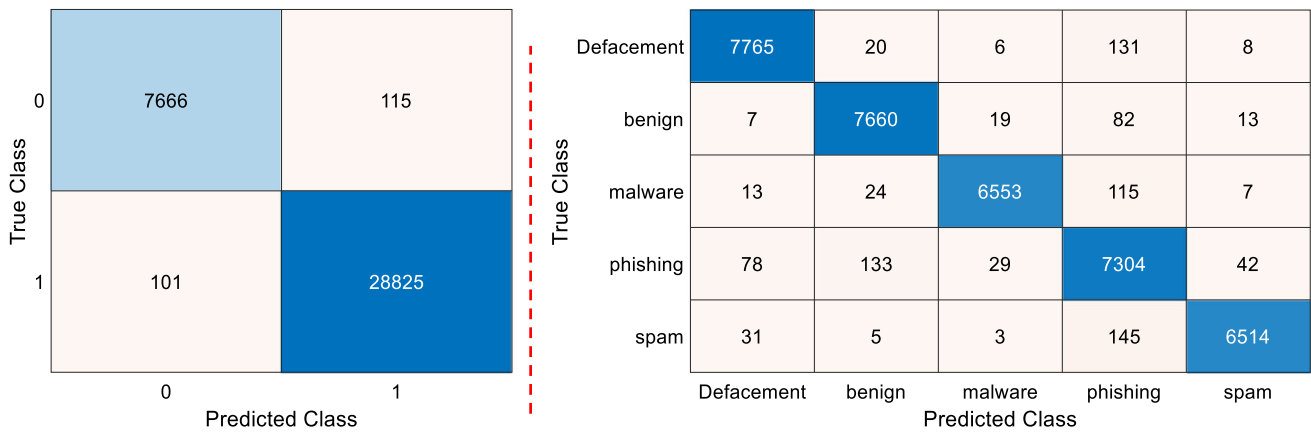
## 5.1 Results and Analysis

Table 3 and Fig. 7 provide a summary of the experimental results for the performance evaluation for the proposed system using four learning models: model 1(En\_Bag), model 2 (En\_kNN), and model 3 (En\_Bos), and model 4 (En\_Dsc). The comparative study considers the detection/classification accuracy, the detection/classification precision, the detection/classification sensitivity (recall), the detection/classification harmonic mean (F Score), and the detection/classification time. Our empirical assessment reveals that the ensemble of bagging trees model (En\_Bag) provides better performance rates than other ensemble models for detection and classification tasks. On the other hand, the ensemble of k-nearest neighbors (En\_KNN) methods provide the highest inference speed compared with other ensemble models for both detection and classification tasks, confirming their appropriateness for high-bandwidth networks. Similarly, the ensemble of boosted trees (En\_Bos) provided competitive performance rates with low detection time. The lowest performance rates were reported for the ensemble of subspace discriminator (En\_Dsc) scoring accuracy that falls below the best model by 10–20%. Overall, *En\_Bag* model provided the best outcomes in terms of all performance indicators along with a very low detection/classification time ( $6.67\mu\text{Sec} - 11.77\mu\text{Sec}$ ) that makes it suitable for real-time communication of Internet of things (IoT) and Cyber-Physical Systems (CPS) networks [43, 46].

## 5.2 Further Investigation for the best model

In this subsection, we focus on analyzing further results for the best-extracted model, the En\_Bag model. Figure 8 shows the confusion matrix of the En\_Bag based system for (a) the 2-classes detection model [0 for benign and 1 for anomaly] and (b) the multi-classes classification model (considering five classes benign, spam, phishing, malware, and defacement). According to the detection matrix, it can be clearly seen that most samples are correctly classified as benign (TP= 7666 and FP = 115) and malicious (TN =28825 and FN = 101). Similarly, for multi-class classification, the majority of samples are correctly classified (blue diagonal) with slight differences in the confusion rates between the different classes. One major observation is that classes for the multi-class classifier are almost perfectly balanced regarding the number of samples per class. At the same time, the classes for the two-class classifier are imbalanced in terms of the number of samples per class. Thus, ensemble learning is the optimal learning and sampling technique for balanced and imbalanced data [47].

Also, Table 4 and Fig. 9 present the summary of experimental results for the detection system using the ensemble of bagging trees: Performance evaluation metrics for the binary classes and the overall. The table reveals the performance metrics for individual classes and the overall metrics for the detection-based En\_Bag model. Accordingly, the performance results for the “anomaly” class are slightly higher than their peers for the “Benign” class. This can be justified by the extremely larger number of samples



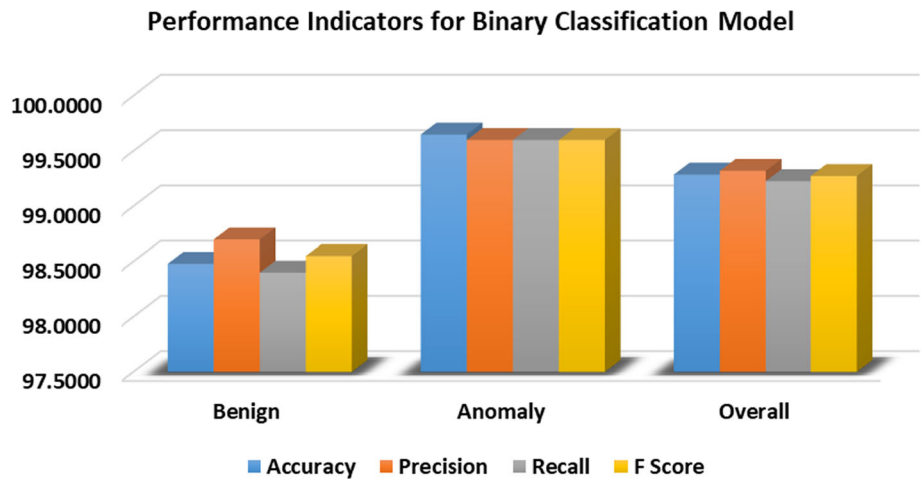
**Fig. 8** Confusion Matrix of En\_Bag Based System for **a** 2-Classes Detection Model [0 for benign and 1 for anomaly] **b** 5-Classes Classification Model

**Table 4** Summary of experimental results of the detection system using the ensemble of bagging trees: Performance evaluation metrics for the binary classes and the overall

Metric	Benign	Anomaly	Overall
Accuracy	98.48 (± 0.07)	99.65 (± 0.09)	99.30 (± 0.11)
Precision	98.70 (± 0.06)	99.60 (± 0.06)	99.32 (± 0.05)
Recall	98.40 (± 0.05)	99.60 (± 0.05)	99.23 (± 0.07)
F Score	98.55 (± 0.06)	99.60 (± 0.06)	99.27 (± 0.06)

trained by the model for the “anomaly” class than that for the “Benign” class, which in turn, improved the detectability for the “anomaly” class in terms of the different evaluation metrics, especially for the detection accuracy metric. Overall, the system exhibited extreme performance indicators scoring an average of 99.30% for detection accuracy, precision, recall, and the F score. Such results are very attractive and competitive; thus, the model can be deployed effectively to uncover malicious URLs for different web applications.

**Fig. 9** Summary of experimental results of the detection system using the ensemble of bagging trees: Performance evaluation metrics for the binary classes and the overall

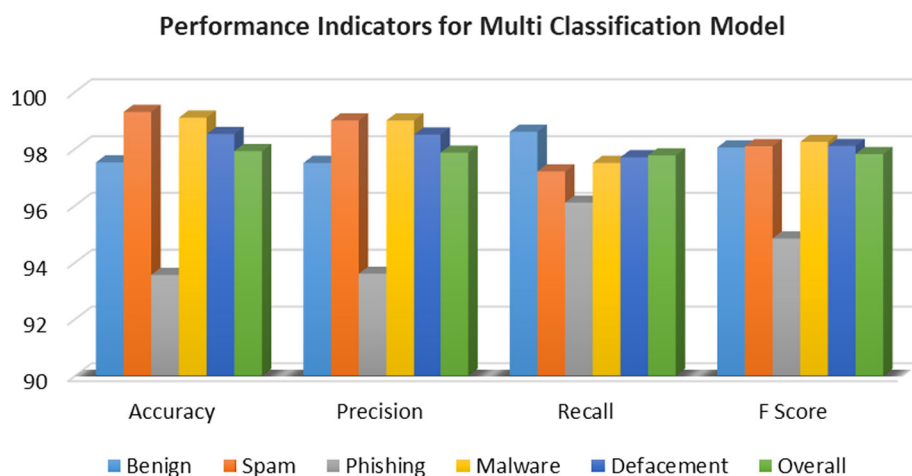


**Table 5** Summary of experimental results of the classification system using the ensemble of bagging trees: Performance evaluation metrics for the individual classes and the overall

Metric	Benign	Spam	Phishing	Malware	Defacement	Overall
Accuracy	97.52 (± 0.07)	99.30 (± 0.01)	93.56 (± 0.07)	99.10 (± 0.03)	98.52 (± 0.03)	97.92 (± 0.03)
Precision	97.51 (± 0.07)	99.03 (± 0.03)	93.60 (± 0.05)	99.00 (± 0.09)	98.50 (± 0.07)	97.87 (± 0.04)
Recall	98.62 (± 0.05)	97.20 (± 0.07)	96.10 (± 0.03)	97.5 (± 0.05)	97.70 (± 0.09)	97.77 (± 0.03)
F Score	98.05 (± 0.06)	98.09 (± 0.05)	94.84 (± 0.04)	98.25 (± 0.07)	98.10 (± 0.08)	97.82 (± 0.04)



**Fig. 10** Summary of experimental results of the detection system using the ensemble of bagging trees: Performance evaluation metrics for the binary classes and the overall



Moreover, Table 5 and Fig. 10 present the summary of experimental results for the classification system using the ensemble of bagging trees: Performance evaluation metrics for the multi-classifier and the overall. The table reveals the performance metrics for individual classes (Benign, Spam, Phishing, Malware, and Defacement) and the overall metrics for the classification-based En\_Bag model. Accordingly, the performance results for the “Spam” and “Malware” classes have recorded the highest accuracy and precision rates with average rates of 99%, slightly higher than their peers for the same metrics. On the other hand, the system has recorded the highest sensitivity toward the “Benign” class scoring a recall rate of 98.6%, marginally higher than its peers for the same metric. However, the harmonic mean (harmonic average of recall and precision, i.e., F Score) was almost the same for all classes with an average rate of 98.1% except for the “Phishing” class, which recorded the lowest (relatively lowest) performance metrics among all classes. This can be observed through the confusion matrix analysis, which reflects that the “Phishing” class has conquered the largest false rates (FP,

FN), among others. Overall, the system exhibited extreme performance indicators scoring an average of 97.8% for classification accuracy, precision, recall, and the F score. Such results are very attractive and competitive. Thus, the model can be deployed effectively to uncover the malicious URLs for different web applications and provide a further classification for the target malware URL class.

Eventually, Table 6 provides a performance comparison of our proposed malicious URL detection and classification system with other up-to-date state-of-art malicious URL detection and/or classification systems developed via machine learning (ML) or deep learning (DL) models. The table differentiates our best experiential outcomes attained for the En\_Bag-based model from the corresponding outcomes stated in the existing state-of-art models. In addition to the year of publication, the assessment in the table evaluates five analogous design performance extents, comprising the detection scheme (using the ML or DL model), the modeling task (detection for two-class classifiers or classification for multi-class classifiers), the number of target classes (2 for the detection or > 2 for

**Table 6** Comparison with other state-of-the-art models

Reference	Year	Task	Model	#. Of Classes	Accuracy	F1 Score
Ibrahim et al. [41]	2017	Detection	RFC_FS	2-Classes	95.20%	Not_Rep
Subasi et al. [42]	2017	Detection	RFC	2-Classes	97.30%	97.40%
Peng et al. [43]	2018	Detection	MNB	2-Classes	86.61%	92.96
Patil et al. [43]	2018	Detection	RFC	2-Classes	96.58%	93.64%
Zamir et al. [44]	2019	Detection	KNN-RF-Bag	2-Classes	97.40%	96.80%
Saha et al. [21]	2020	Classification	MLP	3-Classes	93.00%	Not_Rep
Odeh et al. [22]	2020	Detection	MLP	2-Classes	98.50%	97.30%
Barlow et al. [45]	2020	Classification	MB_CNN	6-Classes	94.16%	91.30
Al-Haija et al. [26]	2021	Detection	En_ODT	2-Classes	97.40%	96.30%
Rana et al. [30]	2022	Detection	LMT	2-Classes	96.90%	96.60%
Proposed	2022	Detection	En_Bag	2-Classes	99.30%	99.27%
Proposed	2022	Classification	En_Bag	5-Classes	97.90%	97.82%

classification), the number of the testing accuracy rates for the detection systems, and the harmonic mean (F score) rates for the detection systems.

Successively, the table contemplates ten different malicious URL detection systems developed during the last five years (from 2017 to 2022), in conjunction with our proposed in- malicious URL detection and classification system (which depends on the ensemble of bagging trees (En\_Bag) as the principal learning model for our system). The reported detection systems encompass the following supervised learning techniques the random forest classifier with feature selection (RFC\_FS) employed by Ibrahim et al. [41], the random forest classifier (RFC) used by Subasi et al. [42], the Multinomial Naïve Bayes (MNB) utilized by Peng et al. [43], the random forest classifier (RFC) applied by Patil et al. [43], the hybrid classifier model of K-nearest neighbors, random forest, and bagging trees (KNN-RF-Bag) implemented by Zamir et al. [44], the multilayer perceptron (MLP) used by Saha et al. [21], the multilayer perceptron (MLP) used by Odeh et al. [22], MobileNet convolutional neural network (MB\_CNN) utilized by Barlow et al. [45], Ensemble of optimizable decision trees (En\_ODT) used by Abu Al-Haija et al. [26], and logistic model trees (LMT) employed by Rana et al. [30].

Rendering the discussion presented above and the results provided in Table 6, one can presume that our malicious detection and classification system is outstanding due to the highest performance records over the other compared state-of-the-art schemes. The proposed model has enhanced the validation accuracy by 1.9 – 12.7% over the evaluated models. In addition, the proposed system can be productively implemented for real-time ecosystems due to the low prediction delay required by the proposed system; the intelligent model needs only 6.67  $\mu$ Sec and 11.77 $\mu$ Sec to provide the detection and classification outcomes, respectively.

## 6 Conclusions and remarks

This paper discusses a new detection and classification system to identify and categorize the malicious uniform resource locators (URLs), which is developed, trained, validated, evaluated, and discussed. The proposed system makes use of four ensembles supervised machine learning approaches, namely the ensemble of bagging trees (En\_Bag) approach, the ensemble of k-nearest neighbor (En\_kNN) approach, the ensemble of boosted decision trees (En\_Bos) approach, and the ensemble of subspace discriminator (En\_Dsc) approach. The implemented models were evaluated on the ISCX-URL2016 dataset, a modern and global multi-class dataset for malicious URLs. Firstly, we identify the URLs as either benign or malware

using a binary classifier. Secondly, we classify the URL classes based on their feature into five classes: benign, spam, phishing, malware, and defacement. Five performance indicators were evaluated and compared for the different models, including accuracy, precision, recall, F Score, and detection time, to characterize the models. The simulation results showed that the En\_Bag approach reported the best-of-all performance rates scoring accuracy rates of 99.3% and 97.92% for the 2-class and 5-class classifications, respectively. Besides, the comparison with existing state-of-the-art models showed that our best results are better than any prior.

**Author contributions** Conceptualization was contributed by QAAH.; methodology was contributed by QAAH.; software was contributed by QAAH.; validation was contributed by QAAH., and MF.; formal analysis was contributed by QAAH.; investigation was contributed by QAAH., and MF.; resources were contributed by QAAH.; data curation was contributed by QAAH., and MF.; writing—original draft preparation, was contributed by QAAH., and MF.; writing—review and editing, was contributed by QAAH., and MF.; visualization was contributed by QAAH., and M.F.; supervision was contributed by QA.AH.; funding acquisition was contributed by QAAH., and MF. All authors have read and agreed to the published version of the manuscript.

**Funding** This research received no external funding.

**Data availability** The data associated with this research can be retrieved from Canadian Institute for Cybersecurity (<https://www.unb.ca/>).

## Declarations

**Conflicts of interest** The authors declare no conflict of interest.

## References

1. Key Internet Statistics to Know in (2022) (Including Mobile) - BroadbandSearch.” [Online]. Available: <https://www.broadbandsearch.net/blog/internet-statistics>. Accessed: 04-Aug-2022.
2. Reports PE, Trends PS, Measurement BP, Attacks, Targeted M, Sectors I, Quarter, (2022) [Online]. Available: [https://docs.apwg.org/reports/apwg\\_trends\\_report\\_q1\\_2022.pdf](https://docs.apwg.org/reports/apwg_trends_report_q1_2022.pdf). Accessed: 23-Jun-2022.
3. Saleem Raja A, Vinodini R, Kavitha A (2021) Lexical features based malicious URL detection using machine learning techniques. Mater Today: Proceed 47:163–166
4. Al-Fayoumi M, Alwidian J, Abusaif M, East IM (2020) Intelligent Association Classification Technique for Phishing Website Detection. Int Arab J Inform Technol 17(4):163
5. Haynes K, Shirazi H, Ray I (2021) Lightweight URL-based phishing detection using natural language processing transformers for mobile devices. Procedia Computer Science 191:127–134
6. R. Abdulraheem, A. Odeh, M. Al Fayoumi and I. Keshta, “Efficient Email phishing detection using Machine learning,” 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 2022,

- pp. 0354–0358. <https://doi.org/10.1109/CCWC54503.2022.9720818>
7. Yaseen YA, Qasaimeh M, Al-Qassas RS, Al-Fayoumi M (2021) E-mail fraud attack detection using hybrid machine learning approach. *Recent Adv Comput Sci Commun.* 14(5):1370–1380
  8. Zahra SR, Chishti MA, Baba AI, Wu F (2022) Detecting covid-19 chaos driven phishing/malicious URL attacks by a fuzzy logic and data mining based intelligence system. *Egyptian Inform J* 23(2):197–214
  9. Sayamber AB, Dixit AM (2014) Malicious URL detection and identification. *Int J Comput Appl* 99(17):17–23
  10. Ibrahim DR and Hadi AH (2017) Phishing websites prediction using classification techniques, in: *Proceedings - 2017 International conference on new trends in computing sciences, ICTCS 2017, 2018-Janua*, 133–137
  11. Subasi A, Molah E, Almkallawi F, Chaudhery TJ (2017) Intelligent phishing website detection using random forest classifier, in: *2017 International conference on electrical and computing technologies and applications, ICECTA 2017, 2018-Janua*, pp. 1–5,
  12. Peng T, Harris I, Sawa Y (2018) Detecting phishing attacks using natural language processing and machine learning, in: *Proceedings - 12th IEEE international conference on semantic computing, ICSC 2018, 2018-Janua*, pp. 300–301
  13. Patil V, Thakkar P, Shah C, Bhat T, Godse SP (2018) Detection and prevention of phishing websites using machine learning Approach, in: *Proceedings - 2018 4th international conference on computing, communication control and automation, ICCUBEA 2018*, pp. 2–6
  14. Patil D, Patil J (2018) Feature-based malicious url and attack type detection using multi-class classification. *The ISC Int J Inform Secur* 10(2):141–162
  15. Adebowale MA, Lwin KT, Sánchez E, Hossain MA (2019) Intelligent web-phishing detection and protection scheme using integrated features of Images, frames and text. *Expert Syst Appl* 115:300–313
  16. Yi P, Guan Y, Zou F, Yao Y, Wang W, Zhu T (2018) Web phishing detection using a deep learning framework. *Wireless Commun Mobile Comput.* <https://doi.org/10.1155/2018/4678746>
  17. Zamir A et al (2020) Phishing web site detection using diverse machine learning algorithms. *Electronic Library* 38(1):65–80
  18. Sahingoz OK, Buber E, Demir O, Diri B (2019) Machine learning based phishing detection from URLs. *Expert Syst Appl* 117:345–357
  19. Varaprasada Rao P, Govinda Rao S, Chandrasekhar Reddy P, Anil Kumar BS, Anil Kumar G (2019) Detection of malicious uniform resource locator. *Int J Recent Technol Eng* 8:41–47
  20. Dsouza, Rohan. Malicious URL (s) classification. Diss. Dublin, National College of Ireland, 2020. <https://norma.ncirl.ie/4138/1/rohansouza.pdf>
  21. Saha I, Sarma D, Chakma RJ, Alam MN, Sultana A, Hossain S (2020) Phishing attacks detection using deep learning approach, in: *Proceedings of the 3rd international conference on smart systems and inventive technology, ICSSIT 2020, Icssit*, pp. 1180–1185
  22. Odeh A, Keshta I, Abdelfattah E (2020) Efficient detection of phishing websites using multilayer perceptron. *Int J Interact Mobile Technol* 14(11):22–31
  23. Barlow L, Bendiab G, Shiaeles S, Savage N (2020) A novel approach to detect phishing attacks using binary visualisation and machine learning, in: *proceedings - 2020 IEEE world congress on services, SERVICES 2020*, pp. 177–182
  24. Xiao X, Zhang D, Hu G, Jiang Y, Xia S (2020) CNN-MHSA: a convolutional neural network and multi-head self-attention combined approach for detecting phishing websites. *Neural Netw* 125:303–312
  25. Shirazi H, Muramudalige SR, Ray I, and Jayasumana AP (2020) Improved Phishing Detection Algorithms using Adversarial Autoencoder Synthesized Data, in: *proceedings - conference on local computer networks, LCN*, vol. 2020-Novem, pp. 24–32
  26. Al-Haija QA, Badawi AA (2021) URL-based phishing websites detection via machine learning. *Int Conf Data Anal Business Ind (ICDABI) 2021:644–649.* <https://doi.org/10.1109/ICDABI53623.2021.9655851>
  27. Maini A, Kakwani N, Ranjitha B, Shreya MK, Bharathi R (2021) Improving the performance of semantic-based phishing detection system through ensemble learning method, in: *2021 IEEE Mysore Sub Section international conference, MysuruCon 2021*, pp. 463–469
  28. Wang Z, Ren X, Li S, Wang B, Zhang J, Yang T (2021) A malicious URL detection model based on convolutional neural network. *Secur Netw.* <https://doi.org/10.1155/2021/5518528>
  29. Manoj P, Bhuvan Kumar Y, Rakshitha D, Megha G (2021) Detection and classification of phishing websites. *Trends Comput Sci Inform Technol* 6:053–059
  30. Abdurraheem R, Odeh A, Al Fayoumi M, Keshta I (2022) Efficient e-mail phishing detection using machine learning, in: *2022 IEEE 12th annual computing and communication workshop and conference, CCWC 2022*, pp. 354–358
  31. URL dataset, by Canadian Institute for Cybersecurity (CIC). <https://www.unb.ca/cic/datasets/url-2016.html>
  32. Al-Haija QA, Gharaibeh M, Odeh A (2022) Detection in adverse weather conditions for autonomous vehicles via deep learning. *AI* 3:303–331
  33. Huang S, Cai N, Pacheco PP, Narrandes S, Wang Y et al (2018) Applications of support vector machine (SVM) learning in cancer genomics. *Cancer Genom - Proteom* 15(1):41–51
  34. Lubis AR, Lubis M (2020) Optimization of distance formula in k-nearest neighbor method,. *Bullet Electrical Eng Inform* 9(1):326–338
  35. Abu Al-Haija Q, Al-Saraiheh J (2022) Asymmetric identification model for human-robot contacts via supervised learning. *Symmetry* 14:591. <https://doi.org/10.3390/sym14030591>
  36. Abu Al-Haija Q (2022) Top-down machine learning-based architecture for cyberattacks identification and classification in IoT communication networks. *Front Big Data* 4:782902
  37. Tsogbaatar E, Bhuyan MH, Taenaka Y, Fall D, Gonchigsumlaa K, Elmroth E, Kadobayashi Y (2021) DeL-IoT: A deep ensemble learning approach to uncover anomalies in IoT. *Int Things* 14:10031
  38. Zidi S, Mihoub A, Qaisar SM, Krichen M, Al-Haija QA (2022) Theft detection dataset for benchmarking and machine learning based classification in a smart grid environment. *J King Saud Univ-Comput Inform Sc.* 35:13
  39. Albulayhi K et al (2022) IoT intrusion detection using machine learning with a novel high performing feature selection method. *Appl Sci* 12:5015. <https://doi.org/10.3390/app1210501>
  40. Abu Al-Haija Q, Al-Badawi A (2022) Attack-aware IoT network traffic routing leveraging ensemble learning. *Sensors* 22:241. <https://doi.org/10.3390/s22010241>
  41. Ibrahim DR, Hadi, AH (2017) Phishing websites prediction using classification techniques, in: *International Conference on New Trends in Computing Sciences (ICTCS'17)*, IEEE, pp.133–137
  42. Subasi A, Molah E, Almkallawi, F and Chaudhery, TJ (2017) Intelligent phishing website detection using random Forest classifier, *International conference on electrical and computing technologies and applications (ICECTA '17)*, IEEE, pp.1–5
  43. Peng T, Harris I, Sawa Y (2018) Detecting phishing attacks using natural language processing and machine learning, in: *2018 IEEE 12th inter. conference on semantic computing (ICSC)*, 2018, 300–301, <https://doi.org/10.1109/ICSC.2018.00056>

44. Zamir A, Khan HU, Iqbal T, Yousaf N, Aslam F, Anjum A, Hamdani M (2019) Phishing web site detection using diverse machine learning algorithms. *Electron Libr* 38(1):65–80. <https://doi.org/10.1108/EL-05-2019-0118>
45. Barlow L, Bendiab G, Shiaeles S, Savage N (2020) A novel approach to detect phishing attacks using binary visualisation and machine learning. *IEEE World Congress on Serv (SERVICES) 2020*:177–182. <https://doi.org/10.1109/SERVICES48979.2020.00046>
46. Al-Haija QA, McCurry CD, Zein-Sabatto S (2020) A real time node connectivity algorithm for synchronous cyber physical and IoT network systems. *SoutheastCon 2020*:1–8. <https://doi.org/10.1109/SoutheastCon44009.2020.9249730>
47. Tyagi S, and Mittal S (2020) Sampling approaches for imbalanced data classification problem in machine learning. *Proceedings of ICRIC 2019*. Springer, Cham, 2020. 209–221.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.