JCBI

Journal of Computing & Biomedical Informatics
ISSN: 2710 - 1606

Volume 07 Issue 02
2024

Research Article
https://doi.org/10.56979/702/2024

Optimizing Malicious Website Detection with the XGBoost Machine Learning Approach

Fazal Malik[1*], Muhammad Suliman[1], Muhammad Qasim Khan[1], Noor Rahman[2,] Khairullah Khan[3], Muhammad Khan[1]

[1]Department of Computer Science, Iqra National University Peshawar, Khyber Pakhtunkhwa (KPK), Pakistan.
[2]Department of Computer science and Engineering, AL- Fayha College, 6480 Al Fayha, Al Jubayl 31961, Saudi Arabia.
[3]Department of Computer Science, University of Science & Technology, Bannu, Khyber Pakhtunkhwa (KPK), Pakistan.
[*]Corresponding Author: Fazal Malik. Email: fazal.malik@inu.edu.pk

Received: April 22, 2024 **Accepted:** August 01, 2024 **Published:** September 01, 2024

Abstract: The rising threat of malicious websites demands advanced detection methods for robust cybersecurity. Traditional approaches, such as rule-based systems and machine learning models like Random Forest and Support Vector Machine (SVM), often struggle to balance precision and recall. This research introduces an innovative methodology using the XGBoost algorithm to detect malicious URLs. The study follows a four-step approach: (1) Dataset Acquisition—utilizing the "Malicious Website URLs" dataset from Kaggle; (2) Data Preprocessing—including data cleaning, feature selection, and transformation to optimize model training; (3) Model Implementation—applying XGBoost, an ensemble learning algorithm known for its superior performance, to train the model on the preprocessed dataset; and (4) Model Evaluation—assessing performance through metrics such as accuracy, precision, recall, and F1-score. The results show that XGBoost achieves 88.89% precision and 86.6% accuracy, outperforming conventional methods and offering a balanced trade-off between precision and recall. This research highlights the significance of precise feature selection and model optimization, reducing human intervention and enhancing cybersecurity defenses. The findings demonstrate XGBoost's effectiveness in minimizing false positives and negatives, making it a valuable addition to existing cybersecurity frameworks. This study underscores the critical role of advanced machine learning techniques and accurate feature selection in strengthening defenses against evolving cyber threats.

Keywords: Malicious Websites; Cyber Security; Types of Malicious Entities; XGBoost Algorithm; Prediction.

1. Introduction

In the field of cyber security, the rise of malicious websites poses a continuous and changing danger to global computer networks. These platforms use cunning strategies to trick unsuspecting users into compromising the systems, which introduces serious security holes. As such, the development of strong detection systems that can quickly identify and neutralize a broad range of cyberthreats is imperative [1].

In response to changing cyber security risks, this study investigates the use of machine learning (ML) to enhance the detection of rogue websites. By enabling proactive defense against new threats, artificial intelligence (AI) integration improves cyber security. With machine learning, such as XGBoost, human intervention is reduced while detection efficiency and accuracy are increased [2]. Early detection of phishing URLs is crucial. A study is presented that detects anomalous behaviors in URLs using supervised learning techniques like Random Forest (RF) and Support Vector Machine (SVM). The proposed method is based on tracking and collecting data from ongoing attack activities instead of depending on outdated datasets [3].

Recognizing phishing URLs at an early stage is crucial. This paper presents a study that detects anomalous behaviors in URLs using supervised learning techniques like Support Vector Machine (SVM) and Random Forest (RF). Instead of depending on outdated statistics, the proposed system tracks and collects data from ongoing campaigns of attacks [4]. Some of the issues with current detection methodologies include limited or outdated feature sets, short datasets, insufficient advanced evaluation, and lengthy feature extraction times for content-based, heuristic, AI, and third-party methods. A machine learning framework using Random Forest, Support Vector Machine, K-Nearest Neighbors, Multilayer Perceptron, Naive Bayes, and Logistic Regression is presented to identify phishing URLs based on URL components and properties [5].

A range of machine learning techniques are employed in a study to effectively and accurately thwart phishing assaults. Among these algorithms are a hybrid LSD model, Decision Tree, Linear Regression, Random Forest, Naive Bayes, Gradient Boosting Classifier, K-Neighbors Classifier, and Support Vector Classifier [6]. Four models that employ decision trees (DTs), random forests (RF), support vector machines (SVMs), and artificial neural networks (ANNs) are developed and evaluated using the UCI phishing domains dataset [7]. The rapid emergence of new threats has made it challenging for classic blacklist-based phishing detection algorithms to predict phishing websites. An approach to address this issue was provided by combining deep neural networks (DNN) and variational autoencoders (VAE) in a deep learning-based phishing detection technique [8].

The Random Forest Classifier and Convolutional Neural Network (CNN) are two examples of Deep Learning and Machine Learning classifiers that are used in a proposed study to evaluate multiclass malicious URL detection. Experimental results show that the recommended characteristics and actions improve the capacity to recognize risky URLs [9]. To classify websites as safe or risky, a study makes use of neural networks, multiple Naive Bayes techniques, and logistic regression. Naive Bayes outperformed other methods in terms of output. To precisely identify websites that are secure and vulnerable, this methodology was improved [10]. Viruses are installed via malicious websites, which also interfere with normal processes and steal data by downloading ostensibly necessary files like video codecs. Users routinely come across malware, spam, and phishing even with safeguards in place. This study promotes a strong detection system that blocks threats proactively without content inspection by using URL (Uniform Resource Locator) data and learning methodologies [11, 12]. Unauthorized access to systems is gained by malware, which includes email viruses, worms, Trojan horses, ransomware, spyware, logic bombs, backdoors, rootkits, and logic bombs. Worms replicate themselves, spyware collects data, and viruses propagate across files. Vulnerabilities are exploited by email viruses, Trojan horses, logic bombs, ransomware, backdoors, rootkits, and keyloggers. Similar to weather forecasting, prediction makes use of past data to predict future events [13].

Researchers discover malicious URLs by applying data mining techniques on a training set of 6,000 samples and the big data RIPPER (Repeated Incremental Pruning to Produce Error Reduction) algorithm [14] . Online threats are addressed using a machine learning (ML)-driven strategy that prioritizes dynamic security in order to thwart phishing. SVMs incorporate harmful URLs that are found in order to prevent user-end assaults [15]. The majority of malicious queries are successfully isolated before they reach resolved IP addresses thanks to an ML packet module included in bundled transmission designs for DNS attack detection. To locate and resolve such bottlenecks, the methodology makes use of standard machine learning libraries and the DPDK (Data Plane Development Kit) [16]. By predicting insider threats, AI (Artificial Intelligence) classifiers on the CERT (Computer Emergency Response Team) dataset achieve a Meta classifier under the ROC (Receiver Operating Characteristic) curve. Preprocessing official log data, representing customer failures, and improving classifier accuracy through thorough assessments are the main goals [17].

With no need for complex selection techniques, the study demonstrated the effectiveness of many classifiers by approaching the problem of matching dangerous URLs as a pairwise arrangement and achieving a high degree of accuracy [18]. Introducing MalNet, a cutting-edge malware detection solution that uses opcode groupings and grayscale photos to train CNN (Convolutional Neural Network) and LSTM (Long Short-Term Memory) systems [19]. After a thorough analysis of AI-driven dangerous URL identification, the authors offered practitioners in digital security and AI analysis deep insights, addressing operational issues and difficulties while providing invaluable assistance for further research [20]. In order to tackle spam, phishing, and malware threats, Naïve Bayes is the best at recognizing harmful URLs from a large dataset utilizing lexical, network, and content factors [21]. When it comes to detecting harmful URLs based on attributes like URL length and site age, logistic regression and support vector machines (SVMs) excel. This technology serves as a safeguard for the expanding Web [22].

Numerous AI-dependent feature computations were used, such as decision trees, Random Forest, Naive Bayes, Support Vector Machines (SVM), Neural Networks, and IBK weak classifiers. These computations' accuracy was assessed and contrasted [23]. Examining resistance-based dangerous URL finding and resolving shortcomings in comprehensiveness and capability to distinguish newly produced URLs, this study used simple algorithms and contrasted results with SVM and LR [24]. In order to improve efficiency and accuracy in URL nature prediction, this study proposes a machine learning strategy that uses Random Forest to detect dangerous URLs [25]. An extensive examination of malware detection tools through information mining techniques was suggested in a study, which also classified malware finding technologies and covered their key elements. Malicious URLs are addressed using machine learning (SVMs, RFs, etc.), which is supported by data reduction methods with instance selection keys for improved model performance [26]. In order to achieve better accuracy without requiring intricate feature selection, the study addressed dual classification problems and demonstrated the efficacy of random forests and multi-layer classifiers [27].

The study addresses a significant unresolved issue in cybersecurity: accurately and efficiently identifying malicious websites. The evolving nature of cyber threats challenges traditional methods, leading to high false positive rates and insufficient protection against new threats.

The primary objective is to develop a robust method that enhances the accuracy and reliability of harmful website identification, reducing human involvement in detection and classification processes to improve efficiency and minimize oversight.

The proposed research employs a four-phase approach for advanced malicious website prediction using the XGBoost algorithm. The methodology includes: (1) Data Preprocessing: Ensuring data is refined for optimal model training and analysis; (2)Data Acquisition: Using a Kaggle dataset to form the basis of the detection model; (3)Detection: Leveraging XGBoost for detecting fraudulent websites due to its superior performance in classification tasks; and (4) Evaluation: Validating model effectiveness through accuracy, precision, recall, and F1-score metrics.

The study's contributions are threefold: (1) it introduces a more balanced detection system, significantly reducing false positives; (2) it improves overall reliability by minimizing trade-offs between recall and precision; (3) it demonstrates the value of optimizing machine learning models, advancing cybersecurity standards and providing practical solutions against emerging threats.

Subsequent sections provide a comprehensive examination of the current state-of-the-art methodologies in the Literature Review section. The Methodology section outlines the proposed framework and approach. The Results and Discussion section presents a detailed analysis of the experimental findings. Finally, the Conclusion section offers a summary of the study and proposes recommendations for future research.

2. Literature Review

Malicious websites pose significant risks to computer systems, including virus infiltration and compromised user security. Ongoing research aims to develop accurate methods for detecting, classifying, and mitigating this cybersecurity threat. This review categorizes existing studies on dangerous website detection for clarity.

An online framework devised for identifying malicious websites using real-time feeds and URL features, surpassing batch processing methodologies [28]. Three objectives were put forth: enhancing a machine learning approach for spotting dangerous online links; training and contrasting algorithms such as Random Forest, Decision Tree, and K-Nearest Neighbor; and determining the most pertinent attributes from which to derive important conclusions [29]. A method proposed to concentrate on customer profiling to detect insider threats through web browsing and email analysis [30]. A data-centric approach was introduced employing deep learning for malware categorization [31].

Machine learning techniques were employed to mitigate user risks within the Chrome browser [32]. Boundary conditions were optimized to counteract malware and minimize adverse effects [33]. An artificial intelligence-driven solution was proposed for classifying malware with minimal computational load [34]. A management rating framework was devised to bolster report accuracy [35]. User connections were forecasted based on content-centric and past social theory data, highlighting equilibrium and status theory [36]. Network efficiency on Windows executable files was assessed using both shallow and deep networks [37]. Intelligent malware discovery approaches were explored, emphasizing extraction and aggregation methods, as well as obstacles in utilizing information mining for detection [38]. Real-time web spam detection was addressed via link-based dispersion methods, focusing on both outgoing and incoming hyperlink spam [39].

By reassigning spasticity scores, a technique was suggested [40] to optimize the spammer ranking algorithm with a focus on collusive group attack aspects. Deep Graph Convolutional Neural Networks (DGCNNs) were suggested as a tool for learning from API call sequences and related behavioral graphs. The first investigation of DGCNNs for behavioral malware detection was made possible by the execution of malware and goodware datasets, which enabled suggested models to reach performance similar to Long-Short Term Memory (LSTM) networks in Area Under the ROC Curve (AUC-ROC) and F1-Score [41]. The comparison of diverse Web spam datasets revealed evolving spam tactics and underscored the importance of reliable filtering techniques [42].

A lightweight setup for detecting harmful Android samples was proposed, utilizing semantic analysis and machine learning [43]. IoT technologies are used by the Consumer Internet of Things (CIoT) to improve everyday convenience. Data from consumer devices has increased dramatically with the swift expansion of the Internet of Things. As information carriers, web pages make CIoT systems vulnerable to security risks associated with spam. In response, a novel classification technique called RFiRF and an intelligent feature extraction method called page2vec are proposed to identify web spam. Page2vec generates different web page attributes by using a score propagation model to calculate goodness and badness scores through web graph links [44]. Because spammers often change the qualities they utilize in their spam emails, traditional techniques of classifying emails become useless over time owing to "Concept Drift." A model is suggested to address this problem and guarantee spam classification for life [45]. Because cell phones are always connected to the internet, they are susceptible to phishing attempts, such as smishing, in which fraudulent SMS messages are delivered to targets. Support Vector Machine (SVM) and Random Forest models were used in an ensemble learning strategy, along with feature extraction techniques including Term Frequency (TF) and Term Frequency-Inverse Document Frequency (TFIDF) [46].

A technique for immediate identification of malicious Java scripts was presented, utilizing machine learning classifiers with dimensionality reduction [47]. A comprehensive survey on malware prognosis was conducted, classifying 89 articles by detection methodologies, identifying hazards, tools, and datasets, laying the foundation for future algorithm development [48]. An ensemble approach inspired by nature is used to improve detection against common online threats via malicious web links. The two datasets used to test the methods were calibrated for each model. It was suggested to use a heterogeneous ensemble and a weighted voting mechanism with weights produced by the Particle Swarm Optimization method. Twelve machine learning models, such as Random Forest, Adaptive Boosting, Support Vector Machine, and Logistic Regression, were compared to create the ensemble [49].

Data mining and AI techniques have been utilized to forecast system infections, demonstrating the ability to detect emerging attack patterns [50]. Enhanced structural integrity post-overhaul was observed, with a caution against excessive training in malicious instances [51]. Research on predicting harmful executable files using behavioral data for endpoint protection has also been conducted [52]. The identification of malicious web pages was explored, proposing an improved network-based learning approach [53].

The relevance of machine learning in identifying phishing attacks and its performance against such threats was examined [54]. A technology for detecting harmful URLs using various features and online AI classifiers was introduced [55]. Common tactics of malicious URL attacks and prevention measures, with a focus on machine learning detection methods, were also investigated [56]. An enhanced XGBoost algorithm was used for classification after the Firefly algorithm was used for feature selection in the development of a model for identifying different hazardous websites. The Particle Swarm Optimization (PSO) algorithm was utilized to attain XGBoost optimization [57].

3. Methodology

In the proposed study, a methodology driven by four core principles is advocated as shown, in Figure 1 and Algorithm 1.

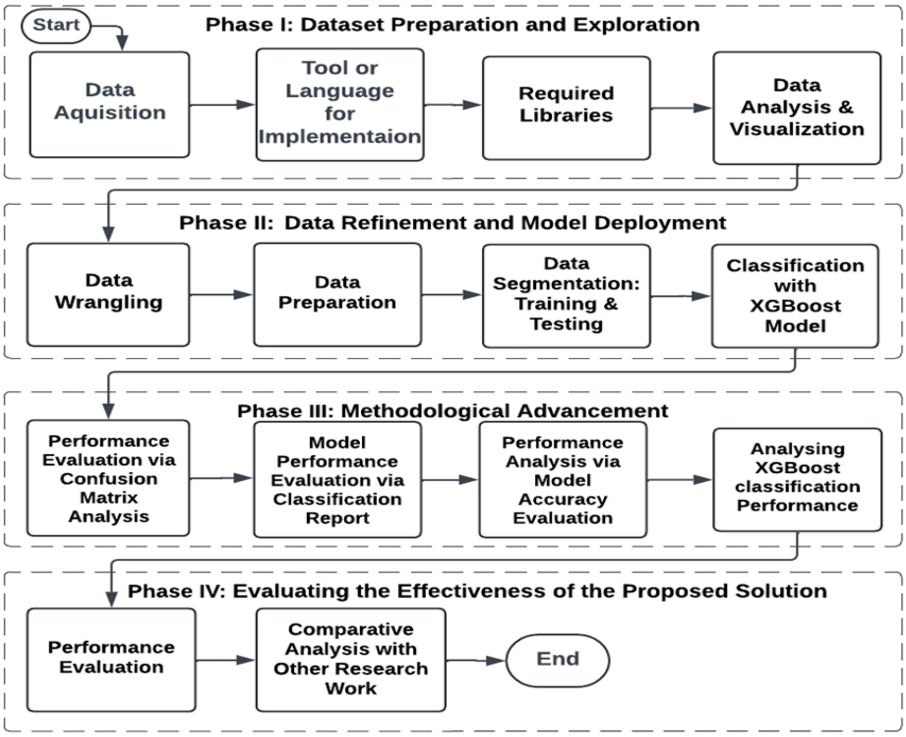
Start
Phase I: Dataset Preparation and Exploration
Data Aquisition
Tool or Language for Implementaion
Required Libraries
Data Analysis & Visualization
Phase II: Data Refinement and Model Deployment
Data Wrangling
Data Preparation
Data Segmentation: Training & Testing
Classification with XGBoost Model
Phase III: Methodological Advancement
Performance Evaluation via Confusion Matrix Analysis
Model Performance Evaluation via Classification Report
Performance Analysis via Model Accuracy Evaluation
Analysing XGBoost classification Performance
Phase IV: Evaluating the Effectiveness of the Proposed Solution
Performance Evaluation
Comparative Analysis with Other Research Work
End

Figure 1. Block Diagram for Proposed Work

```
Algorithm 1: Study on the Prediction of Malicious Websites
Input: Malicious Website Dataset sourced from Kaggle
Output: Findings from the Model Evaluation
Step 1.   Dataset Acquisition
            // Obtain malicious website dataset from Kaggle
    1.1.    Dataset ← Malicious KaggleDataset()
            // Python and Jupyter Notebook as the programming language and tool Selection
    1.2.    Python, Jupyter ← SelectLanguageAndTool()
            // libraries such as Pandas, Numpy, Matplotlib, Math, and Seaborn are imported
    1.3.    Libraries ← ImportLibraries(Pandas, Numpy, Matplotlib, Math, Seaborn)
            // dataset is analyzed using visualization techniques to understand the patterns and characteristics
    1.4.    Visu_Data ← VisualizeData(Dataset, Libraries)
Step 2.   Preprocessing
            // Data wrangling, using scikit-learn's preprocessing libraries to clean for analysis.
    2.1.    preproc_data ← DataWrangling(Dataset)
            // Extract the useful data, refining the dataset to include only the relevant features
    2.2.    refined_dataset ← UsefulData(preproc_data)
            // The refined dataset is split into training and testing sets, identify the dependent and independent classes
    2.3.    train_set, test_set ← SplitDataset(refined_dataset)
  2.3.1. X, Y ← IdentifyClasses(train_set)
  2.3.2. XGBoostClassifier ← InitializeXGBoostClassifier()
  2.3.3. TrainModels(XGBoostClassifier, X, Y)
  2.3.4. optim_models ← OptimizeModels(XGBoostClassifier())
Step 3.   Evaluate model performance
            // confusion matrix is generated to evaluate the performance of the model.
    3.1.    conf_matrix ← EvaluateModels(optim_models, test_set)
            // classification report is produced to assess precision, recall, and F1 scores,
        3.2.    ClassReport ← ClassificationReport(optim_models, test_set)
      3.2.1. metrics ← CalculateMetrics(conf_matrix, precision, recall, F1_score)
              // The predictions made by the models are visualized to compare the
           actual versus predicted outcomes
        3.3.    predictions ← VisualizePredictions(optim_models, test_set)
              // the accuracy score of the predictions is calculated, summarizing the
           overall performance of the model
        3.4.    accuracy_score ← CalculateAccuracyScore(predictions)
```

Firstly, ongoing research endeavors are scrutinized through thorough examinations, using current analysts to identify and eradicate malicious sites. Secondly, a simple approach to forecasting malicious endpoints is introduced. Thirdly, the goal is to reduce reliance on human intervention. Finally, outcomes are assessed and contrasted with prior research, emphasizing precision within the proposed framework.

3.1. Phase I: Dataset Preparation and Exploration

A dataset from Kaggle, containing features of malicious website URLs such as length and character composition is procured to build a predictive model for detecting malicious websites using supervised machine learning techniques.

3.1.1. Dataset Acquisition from Kaggle

The "Malicious Webpages Dataset" from Kaggle, released by Singh and Kumar in 2020 [58], was obtained for the proposed approach. The dataset, produced using MalCrawler, was collected through web scraping between November 2019 and March 2020, ensuring comprehensive global coverage. It has been widely used for developing and testing machine learning (ML) models for malicious website identification. The dataset contains 1,781 instances and 21 features, including URL length, special characters, and other lexical traits, which aid in distinguishing malicious websites. Our ML models are trained and evaluated on this dataset to enhance detection accuracy.

3.1.2. Tool Selection for Implementation

Python is chosen as the primary programming language due to its high-level nature and widespread adoption in data mining, machine learning, and artificial intelligence domains. The decision to use Python was reinforced by its popularity and extensive support within the scientific community. Implementation was carried out using Jupyter Notebook, a versatile platform tailored for scientific problem-solving in data science.

3.1.3. Importing Essential Libraries for Dataset

Key libraries such as Pandas, NumPy, Matplotlib, Seaborn, and Math were imported into Python to facilitate dataset manipulation and analysis. A panda, renowned for its versatility in handling CSV files, was employed for reading and writing datasets. NumPy has augmented analytical capabilities for numerical and matrix data manipulation. Matplotlib and Seaborn were utilized for data visualization, enabling the creation of various graphical representations, such as scatter plots. Additionally, the Math library was utilized for performing mathematical operations, collectively providing comprehensive support for handling both numerical and categorical data.

3.1.4. Exploratory Data Analysis and Visualization

Once the dataset has been imported, we perform a comprehensive analysis to visualize the data and derive insights. The graphical representations are made using the Matplotlib and Seaborn libraries, which facilitate the understanding of dataset features and improve decision-making by offering a concise summary of dataset attributes. Calculations are made to determine parameters such as mean, maximum, and minimum values in descriptive statistics. The dataset, which is divided into float64, int64, and object data types, has 21 columns and 1781 instances. Important statistical data is disclosed, as demonstrated in Table 1 and Table 2, where the mean value for harmful websites is 1781 and the standard deviation is 27.5%.

Table 1. Features of the Malicious Website Dataset

Range index: 1781 entries, 0 to 1780 Data columns (total 21 columns)				
Feature Name	**Description**	**Entries**	**Status**	**Data Type**
URL	The URL of the website.	1781	non-null	Object
URL_LENGTH	The length of the URL.	1781	non-null	int64

NUMBER_SPECIAL_CHARACTERS	The number of special characters in the URL.	1781	non-null	int64
CHARSET	The character set used by the website.	1781	non-null	Object
SERVER	The server type.	1780	non-null	Object
CONTENT_LENGTH	The content length of the response.	969	non-null	float64
WHOIS_COUNTRY	The country listed in the WHOIS information.	1780	non-null	Object
WHOIS_STATEPRO	The state or province listed in the WHOIS information.	1780	non-null	Object
WHOIS_REGDATE	The registration date from WHOIS.	1780	non-null	Object
WHOIS_UPDATED_DATE	The last update date from WHOIS.	1780	non-null	Object
TCP_CONVERSATION_EXCHANGE	The number of TCP conversations exchanged.	1780	non-null	int64
DIST_REMOTE_TCP_PORT	The distance to the remote TCP port.	1780	non-null	int64
REMOTE_IPS	The number of remote IPs.	1780	non-null	int64
APP_BYTES	The number of application bytes exchanged.	1780	non-null	int64
SOURCE_APP_PACKETS	The number of application packets sent from the source.	1780	non-null	int64
REMOTE_APP_PACKETS	The number of application packets sent from the remote end.	1780	non-null	int64
SOURCE_APP_BYTES	The number of application bytes sent from the source.	1780	non-null	int64
REMOTE_APP_BYTES	The number of application bytes received by the remote end.	1780	non-null	int64
APP_PACKETS	The total number of application packets exchanged.	1780	non-null	int64
DNS_QUERY_TIMES	The number of DNS queries made by the URL.	1780	non-null	float64
TYPE	The label indicating whether the URL is malicious (1) or benign (0).	1780	non-null	int64
dtypes: float64(2), int64(12), object(7) memory usage: 292.3+ KB				

Numerous attributes that record the syntactical, lexical, and network-related properties of URLs are included in this collection. In this study, we trained and assessed machine learning models utilizing all 21 characteristics, with particular attention to those that are closely correlated with the structure and behavior of URLs (e.g., URL_LENGTH, NUMBER_SPECIAL_CHARACTERS, and DNS_QUERY_TIMES).

In order to fully capture the range of traits that could help differentiate between malicious and benign websites, we made the decision to include every element in our study. Our models are able to recognize intricate patterns that could be missed if a smaller collection of features were employed thanks to this thorough methodology. After that, the efficacy of utilizing a variety of variables was demonstrated by testing the final prediction model's capacity to correctly identify harmful websites.

Table 2. Statistical Description of Data

	Url_ Length	**Number_ Special_ Characters**	**Content_ Length**	**Tcp_ Conversation_ Exchage**	**Dist_ Remote_ Tcp_port**	**Remote_ Ips**	**App_ Bytes**

count	mean	std	Min	25%	50%	75%	max
1781.000000	56.961258	27.555586	16.000000	39.000000	49.000000	68.000000	249.000000
1781.000000	11.111.735	4.549896	5.000000	8.000000	10.000000	13.000000	43.000000
969.000000	11726.927761	36391.809051	0.000000	324.000000	1853.000000	11323.000000	649263.000000
1781.000000	16.261089	40.500975	0.000000	0.000000	7.000000	22.000000	1194.000000
1781.000000	5.472768	21.807327	0.000000	0.000000	0.000000	5.000000	708.000000
1781.000000	3.060640	3.386975	0.000000	0.000000	2.000000	5.000000	17.000000
1.781000	2.982000	5.605000	0.000000	0.000000	6.720000	2.328000	2.362000

In cyber security research, Python and its libraries provide a strong foundation for methodical analysis. The basis for further stages is laid by this analysis, which makes it easier to apply machine learning models to the prediction of harmful websites.

3.2. Phase II: Data Refinement and Model Deployment

3.2.1. Data Refinement: Ensuring Data Consistency

Data wrangling is the act of methodically removing null, missing, and irrelevant values from datasets using the scikit-learn preprocessing modules. It is essential for data mining and analysis. This stage guarantees clean, well-refined data that is devoid of errors, laying the groundwork for a solid analysis. In research, data wrangling is essential to preserving data integrity. Methods like data collection and missing value identification/resolution are used. The dataset's missing values are shown in Table 3, where the following are identified: server (1 missing value), content_length (812 missing values), and dns_query_times (1 missing value).

Table 3. Visualization Showing Missing Data Patterns

Features	Missing Values
URL	0
URL_LENGTH	0
NUHSER_SPECIAL_CHARACTERS	0
CHARSET	0
SERVER	1
CONTENT_LENGTH	812
WHOIS_COUNTRY	0
WHOIS_STATEPRO	0
WHOIS_REGDATE	0
WHOIS _UPDATED_DATE	0
TCP_CONVERSATION_EXCHAGE	0
DIST_REMOTE_TCP_PORT	0
REMOTE_IPS	0
APP_BYTES	0
SOURCE_APP_PACKETS	0
REMOTE_APP_PACKETS	0
SOURCE _APP_BYTES	0
REMOTE_APP_BYTES	0
APP_ PACKETS	0
DNS_QUERY_TIMES	1
TYPE	0

As seen in Table 4, mean imputation is used to replace these missing data. This methodical procedure ensures that the dataset is clean, which allows for additional analysis.

Table 4. The process for replacing data that are missing

Features	Missing Values
URL	0
URL_LENGTH	0
NUHSER_SPECIAL_CHARACTERS	0
CHARSET	0
SERVER	0
CONTENT_LENGTH	0
WHOIS_COUNTRY	0
WHOIS_STATEPRO	0
WHOIS_REGDATE	0
WHOIS _UPDATED_DATE	0
TCP_CONVERSATION_EXCHAGE	0
DIST_REMOTE_TCP_PORT	0
REMOTE_IPS	0
APP_BYTES	0
SOURCE_APP_PACKETS	0
REMOTE_APP_PACKETS	0
SOURCE _APP_BYTES	0
REMOTE_APP_BYTES	0
APP_ PACKETS	0
DNS_QUERY_TIMES	0
TYPE	0

3.2.2. Data Preparation

After data refinement, the resulting dataset is clean and ready for analysis. Free of irrelevant values, it serves as the foundation for training and testing. This dataset is crucial for the proposed model, helping to distinguish between malicious and non-malicious websites.

3.2.3. Data Segmentation and Division

The dataset is divided into dependent and independent variables, where the dependent variable serves as the target class and the independent variables are the predictors. Using techniques from the sklearn model selection library, the data is further split into training and testing sets. This is achieved by methods such as X_train, X_test, y_train, and y_test [59], ensuring a robust model application.

3.2.4. Machine Learning Classification Model

Classification models are a subset of machine learning algorithms used to categorize data into predefined classes based on data features. These models are trained on labeled datasets to recognize patterns and relationships, allowing them to predict class labels for new data. In the context of identifying potentially malicious websites, various machine learning models, including XGBoost, analyze website features and behaviors to assess risks to users.

- *XGBoost (Extreme Gradient Boosting)*

XGBoost represents an advanced implementation of gradient boosting, renowned for its efficiency and scalability. It enhances traditional gradient-boosting approaches by integrating regularization techniques to mitigate overfitting and leveraging distributed computing for parallelization. Employing decision trees as base learners, XGBoost iteratively constructs an ensemble model by minimizing a specific loss function. Its widespread adoption stems from its ability to achieve cutting-edge performance across diverse classification tasks while remaining computationally efficient.

The XGBClassifier from the xgboost library is employed with predefined parameters, including colsample_bytree, learning_rate, max_depth, alpha, and n_estimators. The integration of the XGBoost model marks a significant stride in attaining heightened accuracy and execution speed, positioning it as a cornerstone in the classification and prediction phase of the research. This methodical approach to data preparation and model implementation enhances the robustness and reliability of the proposed solution for predicting malicious websites using machine learning techniques. Subsequent sections delve into the results obtained and draw conclusions based on the findings.

3.3. Phase III: Methodological Advancement

3.3.1. Performance Evaluation using Confusion Matrix

In this phase, the study introduces a Confusion Matrix to depict the performance of the machine learning model. This matrix compares actual labels with predicted labels, facilitating the calculation of key performance metrics. It is crucial for evaluating the effectiveness of the XGBoost classifier in predicting malicious websites.

The Confusion Matrix highlights classification errors, delineating true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). This breakdown aids in calculating precision, recall, F1-score, and accuracy, providing insight into the model's accuracy and error patterns.

It is essential for diagnosing the classifier's behavior and refining the model's performance in detecting malicious websites.

- *Evaluating XGBoost Model Performance with a Confusion Matrix*

The Confusion Matrix visually represents the performance of a machine learning model by comparing actual and predicted labels. It assesses the effectiveness of the XGBoost classifier in detecting malicious websites and highlights classification errors, providing insights into the algorithm's accuracy and limitations. Figure 2 shows a Confusion Matrix illustrating the XGBoost model's performance.

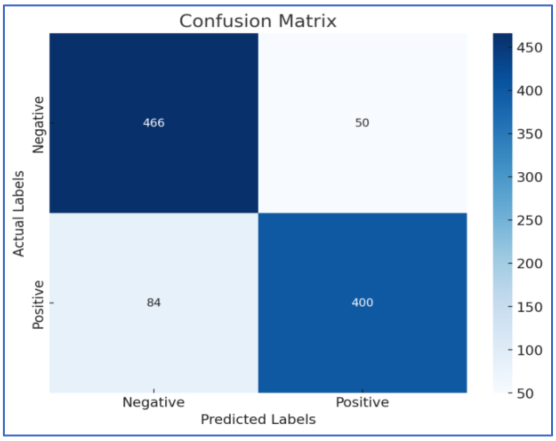

Figure 2. XGBoost Classifier Performance is demonstrated by an example confusion matrix

In the Confusion Matrix, True-Positive (TP), True-Negative (TN), False-Positive (FP), and False-Negative (FN) values reflect the classifier's performance:

- **TP**: Correctly identified malicious websites (400).
- **TN**: Correctly identified non-malicious websites (466).
- **FP:** Non-malicious websites misclassified as malicious (50).
- **FN**: Malicious websites misclassified as non-malicious (84).

These values enable the calculation of performance metrics such as Precision, Recall, F1-Score, and Accuracy, providing a thorough evaluation of the XGBoost classifier's effectiveness in detecting malicious websites and highlighting areas for improvement in cybersecurity.

3.3.2. Model Performance Evaluation through Classification Report

This section evaluates the model's performance through a classification report, following the analysis of the XGBoost Confusion Matrix. The report provides insights into the model's accuracy, precision, recall, and F1 score. The focus is on using machine learning classifiers, specifically XGBoost, to distinguish between malicious and non-malicious websites. Figure 3 presents the performance metrics derived from the XGBoost model.

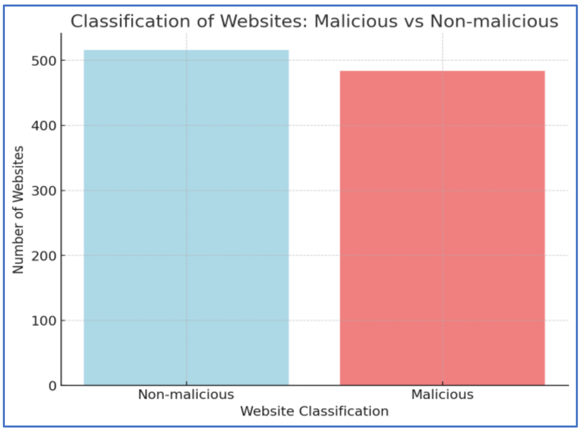

Figure 3. Distribution of Malicious and Non-Malicious Websites as shown by Classification Report

Figure 3 presents the distribution of malicious and non-malicious websites in the dataset. Of the 1000 websites analyzed, 516 are non-malicious, and 484 are malicious. This data serves as a foundation for evaluating the model's accuracy.

Classifying the websites into non-malicious and malicious categories provides key insights into the model's performance. Balancing false positives and false negatives is crucial for refining the model. A high false positive rate can lead to unnecessary alerts, while a high false negative rate may result in missed threats. This data is vital for guiding further model improvements to optimize these trade-offs.

3.3.3. Performance Analysis: Evaluating Model Accuracy

The proposed study evaluates the model's effectiveness by calculating the average accuracy in classifying and predicting malicious websites. The accuracy, derived from the confusion matrix, measures the model's performance by comparing predicted outcomes with actual results. A detailed review of the classification report offers insights into the model's reliability in website classification and prediction.

- *Performance Evaluation Metrics in Malicious Website Prediction*

In the analysis of classification model effectiveness, particularly in predicting malicious websites, diverse evaluation metrics play a pivotal role. This segment undertakes an extensive exploration of these metrics, emphasizing the significance of the accuracy score alongside precision, recall, and the F1 score.

In assessing the effectiveness of a classification model, several metrics play a crucial role. Among them, accuracy (AC) stands out as a fundamental measure, indicating the proportion of correctly predicted instances.

$$Accuracy(AC) = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Precision (PR) and recall (RE) offer insights into the model's ability to accurately classify positive instances and identify all relevant positives, respectively.

$$Precision\ (PR) = \frac{TP}{TP+FP} \quad (2)$$

$$Recall\ (RE) = \frac{TP}{TP+FN} \quad (3)$$

The F1 score provides a balanced measure by considering both precision and recall.

$$F1\ Score = 2 \times \frac{PR \times RE}{PR + RE} \tag{4}$$

These metrics are calculated using formulas (1) to (4) and are essential for understanding and evaluating model performance.

Considering a scenario where 400 instances are correctly classified as malicious (TP), 466 instances as non-malicious (TN), 50 instances as incorrectly classified as malicious (FP), and 84 instances as incorrectly classified as non-malicious (FN), the XGBoost model achieves an overall accuracy of 86.6%. This indicates the model's capability to accurately predict instances across all classes. With precision reported at 88.89%, recall at 82.64% and F1-Score 85.67%, the model demonstrates a high level of accuracy in classifying both positive and negative instances.

3.3.4. Analyzing XGBoost Classifier Performance

The performance evaluation of the XGBoost classifier, summarized in Table 5, highlights its predictive capabilities in identifying malicious websites.

Table 5. The XGBoost Classifier's Performance Measures

Classifiers	Precision	Recall	F1 Score	Accuracy
XGBoost	88.89%	82.64%	85.67%	86.6%

XGBoost achieves a precision of 88.2% and recall of 83.3%, demonstrating strong accuracy in categorizing both positive and negative instances. The F1 score, a balance of precision and recall, is 85.74%, highlighting the model's ability to maintain equilibrium between the two. The average accuracy of 85.6% confirms XGBoost's effectiveness in predicting malicious websites. These metrics provide a robust evaluation of the classifier's performance, underscoring its proficiency in distinguishing between malicious and non-malicious websites. The choice of XGBoost is justified, given its strong performance and adherence to industry standards.

3.3.5. Contribution of XGBoost in Predictive Accuracy Enhancement

This study advances the prediction of malicious websites through performance analysis, focusing on precision, recall, and F1 score metrics, and setting a new standard in cybersecurity research. XGBoost's integration demonstrates the importance of advanced machine learning for complex classification tasks, emphasizing the need for diverse methodologies to improve predictive accuracy.

The comprehensive assessment of XGBoost’s precision, recall, and F1 score provides valuable insights, guiding researchers and industry professionals in developing robust predictive models for cyber threats. These metrics are critical for evaluating model performance, ensuring accurate threat identification while minimizing false positives, which can reduce system trust and drain resources.

Through empirical evidence and methodological rigor, this analysis contributes to the evaluation of predictive models for malicious website detection, supporting future research and strengthening cyber defenses.

4. Results and Discussion

This section thoroughly evaluates the proposed machine learning-driven method for forecasting malicious websites, focusing on its efficacy and dependability. A comparative analysis highlights the superiority of the XGBoost model over previous Random Forest classifiers. The findings demonstrate the approach's potential to enhance cybersecurity, underscoring its relevance for future research and real-world applications.

4.1. Phase IV: Evaluating the Effectiveness of the Proposed Solution

This section evaluates the performance of the machine learning-based approach for predicting malicious websites. The classification and prediction results are analyzed to assess the model's effectiveness and reliability. A comparative analysis with existing research is also performed to evaluate the approach's consistency and coherence.

4.1.1. Performance Evaluation and Comparative Analysis

The results of the proposed study are thoroughly compared to the work by Saeed Ahmad Al Tamimi et al. [22], focusing on the effectiveness of different algorithms in identifying malicious websites, as shown in Table 6.

Table 6. Comparative Analysis of the Proposed Research with Previous Studies

Authors	Data Set	Algorithm	Precision	Recall	F1-Score	Accuracy
Saeed Ahmad Al Tamimi [22]	Malicious and Benign Websites Kaggle	Random Forest	68%	90.7%	77.7%	92%
		SVM	82%	55%	66%	77%
Our proposed research work	Malicious website URLs Dataset from Kaggle	XGBoost	88.89%	82.64%	85.67%	86.6%

Al Tamimi's study utilized the Random Forest and Support Vector Machine (SVM) algorithms on a dataset from Kaggle containing both malicious and benign websites. The Random Forest model achieved a high recall of 90.7% but had a precision of only 68%, resulting in a significant number of false positives. The SVM model showed higher precision (82%) but struggled with a low recall (55%), leading to an overall accuracy of 77%.

In contrast, our proposed research employs the XGBoost algorithm on the Malicious Website URLs dataset from Kaggle, achieving a balanced performance: precision of 88.89%, recall of 82.64%, F1-score of 85.67%, and accuracy of 86.6%. These results demonstrate significant improvements in precision while maintaining competitive recall. This balance is crucial in real-world cybersecurity applications, where false positives can result in unnecessary blocking of benign websites or unwarranted alarms.

The main challenge in malicious website detection is achieving high recall without compromising precision, as this directly affects system reliability. Our XGBoost model addresses this by minimizing false positives while accurately detecting malicious websites, thus improving the overall dependability of the detection system.

Additionally, the scalability and adaptability of the XGBoost model make it suitable for integration into existing cybersecurity frameworks. Its ability to handle large datasets and maintain high performance suggests it can enhance security measures across various platforms. The proposed methodology not only improves accuracy and reliability in malicious URL detection but also sets a new benchmark for algorithm and feature selection in cybersecurity.

This study’s contributions are significant. By adopting XGBoost, we developed a more precise and balanced detection system that outperforms previous methods, particularly in reducing false positives. This advancement in algorithmic choice and feature selection enhances the ability to detect and prevent malicious websites, offering a robust solution for global cybersecurity efforts.

5. Conclusion

This study presents an optimized methodology for detecting malicious websites using the XGBoost algorithm, showing significant improvement over traditional methods like Random Forest and Support Vector Machine (SVM). The XGBoost model achieved an accuracy of 86.6%, with a precision of 88.89% and recall of 82.64%, highlighting its ability to effectively distinguish between malicious and benign URLs. This performance demonstrates the importance of advanced machine learning techniques in addressing complex cybersecurity challenges, where high precision and recall are crucial. The proposed approach offers a better balance between precision and recall compared to existing methods, reducing false positives while maintaining high detection accuracy. This balance is critical in real-world cybersecurity scenarios, where misclassifying benign websites can have significant consequences. Our research makes a valuable contribution to cybersecurity, presenting a thorough performance evaluation of the XGBoost model. The findings offer insights for both researchers and industry professionals, suggesting that integrating the XGBoost algorithm into existing security frameworks can enhance defenses against malicious websites.

While the study's use of a Kaggle dataset is effective, it may not fully represent the variety of real-world cyber threats. Future research should explore more diverse, real-time datasets, consider additional machine learning algorithms, and investigate hybrid models to further improve detection accuracy and adaptability. Developing adaptive cybersecurity measures that incorporate real-time threat detection will be essential for countering emerging threats, laying a solid foundation for continued innovation in the field.

References

1. Reddy Palle, R. "Explore the Application of Predictive Analytics and Machine Learning Algorithms in Identifying and Preventing Cyber Threats and Vulnerabilities within Computer Systems." *International Journal of Science and Research (IJSR)* 12, no. 2 (2023): 1704–1712. https://doi.org/10.21275/es24101104007.
2. Kaur, Ramanpreet, Dušan Gabrijelčič, and Tomaž Klobučar. "Artificial Intelligence for Cybersecurity: Literature Review and Future Research Directions." Information Fusion 97 (2023): 101804.
3. Lam, Nguyen Tung. "Developing a Framework for Detecting Phishing URLs Using Machine Learning." International Journal of Computer Science & Network Security 23, no. 10 (2023): 157–163.
4. Das Guptta, Sumitra, Khandaker Tayef Shahriar, Hamed Alqahtani, Dheyaaldin Alsalman, and Iqbal H. Sarker. "Modeling Hybrid Feature-Based Phishing Websites Detection Using Machine Learning Techniques." Annals of Data Science 11, no. 1 (2024): 217–242.
5. Jalil, Sajjad, Muhammad Usman, and Alvis Fong. "Highly Accurate Phishing URL Detection Based on Machine Learning." Journal of Ambient Intelligence and Humanized Computing 14, no. 7 (2023): 9233–9251.
6. Karim, Abdul, Mobeen Shahroz, Khabib Mustofa, Samir Brahim Belhaouari, and S. Ramana Kumar Joga. "Phishing Detection System through Hybrid Machine Learning Based on URL." IEEE Access 11 (2023): 36805–36822. https://doi.org/10.1109/ACCESS.2023.3252366.
7. Alnemari, Shouq, and Majid Alshammari. "Detecting Phishing Domains Using Machine Learning." Applied Sciences 13, no. 8 (2023): 4649.
8. Prabakaran, Manoj Kumar, Parvathy Meenakshi Sundaram, and Abinaya Devi Chandrasekar. "An Enhanced Deep Learning-Based Phishing Detection Mechanism to Effectively Identify Malicious URLs Using Variational Autoencoders." IET Information Security 17, no. 3 (2023): 423–440.
9. Mumu, Mahmuda Haque, and Tanzina Aishy. "Malicious URL Detection Using Machine Learning and Deep Learning Algorithms." PhD diss., East West University, 2023.
10. Koca, Murat, İsa Avcı, and Mohammed Abdulkareem Shakir Al-hayani. "Classification of Malicious URLs Using Naive Bayes and Genetic Algorithm." Sakarya University Journal of Computer and Information Sciences 6, no. 2 (2023): 80–90. https://doi.org/10.35377/saucis...1273536.
11. Naim, O., D. Cohen, and I. B. Gal. "Ensemble Classification for Stroke Prediction and Diagnosis: Enhancing Accuracy through Collaborative Algorithms." In 2023 International Conference on Research Methodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering (RMKMATE), 2023. https://doi.org/10.1109/rmkmate59243.2023.10368945.
12. Naim, Or, Doron Cohen, and Irad Ben-Gal. "Malicious Website Identification Using Design Attribute Learning." International Journal of Information Security 22, no. 5 (2023): 1207–1217.
13. Jiang, Peng, Jifan Xiao, Ding Li, Hongyi Yu, Yu Bai, Yao Guo, and Xiangqun Chen. "Detecting Malicious Websites from the Perspective of System Provenance Analysis." IEEE Transactions on Dependable and Secure Computing 21, no. 3 (2023): 1406–1423.
14. Thakur, Sonika, Er Meenakshi, and Akansha Priya. "Detection of Malicious URLs in Big Data Using RIPPER Algorithm." In 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 1296–1301. IEEE, 2017.
15. Jeenal, R., G. Preethi, A. Praveena, and A. Preethi. "Malicious URL Detection Using Machine Learning Techniques." International Journal of Computer Applications 8, no. 3 (2019): 1–5.
16. Suga, Toki, Kazuya Okada, and Hiroshi Esaki. "Toward Real-Time Packet Classification for Preventing Malicious Traffic by Machine Learning." In 2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN), 106–111. IEEE, 2019.
17. Hall, Adam James, Nikolaos Pitropakis, William J. Buchanan, and Naghmeh Moradpoor. "Predicting Malicious Insider Threat Scenarios Using Organizational Data and a Heterogeneous Stack-Classifier." In 2018 IEEE International Conference on Big Data (Big Data), 5034–5039. IEEE, 2018.
18. Jiang, Jinfang, Guangjie Han, Feng Wang, Lei Shu, and Mohsen Guizani. "An Efficient Distributed Trust Model for Wireless Sensor Networks." IEEE Transactions on Parallel and Distributed Systems 26, no. 5 (2014): 1228–1237.
19. Yan, Jinpei, Yong Qi, and Qifan Rao. "Detecting Malware with an Ensemble Method Based on Deep Neural Network." Security and Communication Networks 2018 (2018): 7247095.

20. Du, Jun, Erol Gelenbe, Chunxiao Jiang, Haijun Zhang, Yong Ren, and H. Vincent Poor. "Peer Prediction-Based Trustworthiness Evaluation and Trustworthy Service Rating in Social Networks." IEEE Transactions on Information Forensics and Security 14, no. 6 (2018): 1582–1594.
21. Aljabri, Malak, Fahd Alhaidari, Rami Mustafa A. Mohammad, Samiha Mirza, Dina H. Alhamed, Hanan S. Altamimi, and Sara Mhd Bachar Chrouf. "An Assessment of Lexical, Network, and Content-Based Features for Detecting Malicious URLs Using Machine Learning and Deep Learning Models." Computational Intelligence and Neuroscience, 2022, Article 3241216.
22. Al Tamimi, Saeed Ahmad. "Detecting Malicious Websites Using Machine Learning." Journal of Cyber Security Technology 4, no. 1 (2020): 45-60.
23. Islam, Mazharul, and Nihad Karim Chowdhury. "Phishing Websites Detection Using Machine Learning-Based Classification Techniques." In International Conference on Advanced Information and Communication Technology, vol. 10, no. 9, 4393-4402. Chittagong, Bangladesh, 2016.
24. Abdi, Farhan Douksieh, and Lian Wenjuan. "Malicious URL Detection Using Convolutional Neural Network." International Journal of Computer Science, Engineering and Information Technology 7, no. 6 (2017): 1-8.
25. Deebanchakkarawarthi, G., A. S. Parthan, L. Sachin, and A. Surya. "Classification of URL into Malicious or Benign Using Machine Learning Approach." International Journal of Advanced Research in Computer and Communication Engineering 8, no. 2 (2019): 1-4.
26. Abad, Shayan, Hassan Gholamy, and Mohammad Aslani. "Classification of Malicious URLs Using Machine Learning." Sensors 23, no. 18 (2023): 7760.
27. Zhao, Guoshuai, Xueming Qian, and Xing Xie. "User-Service Rating Prediction by Exploring Social Users' Rating Behaviors." IEEE Transactions on Multimedia 18, no. 3 (2016): 496-506.
28. Abu Al-Haija, Qasem, and Mustafa Al-Fayoumi. "An Intelligent Identification and Classification System for Malicious Uniform Resource Locators (URLs)." Neural Computing and Applications 35, no. 23 (2023): 16995-17011.
29. Coste, Claudia-Ioana. "Malicious Web Links Detection: A Comparative Analysis of Machine Learning Algorithms." Studia Universitatis Babeș-Bolyai Informatica (2023): 21-36.
30. Jiang, Jianguo, Jiuming Chen, Kim-Kwang Raymond Choo, Kunying Liu, Chao Liu, Min Yu, and Prasant Mohapatra. "Prediction and Detection of Malicious Insiders' Motivation Based on Sentiment Profile on Webpages and Emails." In MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM), 1-6. IEEE, 2018.
31. Le, Quan, Oisín Boydell, Brian Mac Namee, and Mark Scanlon. "Deep Learning at the Shallow End: Malware Classification for Non-Domain Experts." Digital Investigation 26 (2018): S118-S126.
32. Desai, Anand, Janvi Jatakia, Rohit Naik, and Nataasha Raul. "Malicious Web Content Detection Using Machine Learning." In 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 1432-1436. IEEE, 2017.
33. Ahmad, Ashraf, Yousef S. Abu Hour, and Mahmoud H. DarAssi. "Advance System and Model to Predict Malicious Files Propagation Inside Computer Networks." IET Networks 8, no. 1 (2019): 38-47.
34. Kumar, Ajit, K. S. Kuppusamy, and Gnanasekaran Aghila. "A Learning Model to Detect Maliciousness of Portable Executable Using Integrated Feature Set." Journal of King Saud University-Computer and Information Sciences 31, no. 2 (2019): 252-265.
35. Du, J., C. Jiang, H. Zhang, Y. Ren, and H. V. Poor. "Peer Prediction Based Trustworthiness Evaluation and Trustworthy Service Rating in Social Networks." IEEE Transactions on Network and Service Management (2018). https://doi.org/10.1109/TNSE.2018.2813044.
36. Das, Debasis, and Pushkar Sharma. "Algorithm for Prediction of Negative Links Using Sentiment Analysis in Social Networks." In 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC), 1570-1575. IEEE, 2017.
37. Vinayakumar, R., and K. P. Soman. "DeepMalNet: Evaluating Shallow and Deep Networks for Static PE Malware Detection." ICT Express 4, no. 4 (2018): 255-258.
38. Ye, Yanfang, Tao Li, Donald Adjeroh, and S. Sitharama Iyengar. "A Survey on Malware Detection Using Data Mining Techniques." ACM Computing Surveys (CSUR) 50, no. 3 (2017): 1-40.
39. Egele, Manuel, Clemens Kolbitsch, and Christian Platzer. "Removing Web Spam Links from Search Engine Results." Journal in Computer Virology 7 (2011): 51-62.
40. Zhang, Zheng, Mingyang Zhou, Jun Wan, Kezhong Lu, Guoliang Chen, and Hao Liao. "Spammer Detection via Ranking Aggregation of Group Behavior." Expert Systems with Applications 216 (2023): 119454.

41. de Oliveira, Angelo Schranko, and Renato José Sassi. "Behavioral Malware Detection Using Deep Graph Convolutional Neural Networks." Authorea Preprints (2023).
42. Wang, De, Danesh Irani, and Calton Pu. "Evolutionary Study of Web Spam: Webb Spam Corpus 2011 versus Webb Spam Corpus 2006." In 8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 40-49. IEEE, 2012.
43. Qiu, Junyang, Wei Luo, Lei Pan, Yonghang Tai, Jun Zhang, and Yang Xiang. "Predicting the Impact of Android Malicious Samples via Machine Learning." IEEE Access 7 (2019): 66304-66316.
44. Wang, Rong, Xu Zhuang, Xiaogang Zhu, Ali Kashif Bashir, Maryam M. Al Dabel, and Keping Yu. "Intelligent Web Spam Detection in the Consumer Internet of Things." IEEE Transactions on Consumer Electronics (2024).
45. Mohammad, Rami Mustafa A. "A Lifelong Spam Emails Classification Model." Applied Computing and Informatics 20, no. 1/2 (2024): 35-54.
46. Xu, Hongsheng, Akeel Qadir, and Saima Sadiq. "Enhancing Mobile Cybersecurity: Smishing Detection Using Ensemble Learning and SMOTE." SSRN 4875342 (2024).
47. Khan, Nayeem, Johari Abdullah, and Adnan Shahid Khan. "Defending Malicious Script Attacks Using Machine Learning Classifiers." Wireless Communications and Mobile Computing 2017, no. 1 (2017): 5360472.
48. Kim, Yukyong, Eun-Wha Jhee, Jongwon Choe, Jong-Seok Choi, and Yongtae Shin. "A Measurement Model for Trustworthiness of Information on Social Network Services." In 2015 International Conference on Information Networking (ICOIN), 437-438. IEEE, 2015.
49. Coste, Claudia Ioana. "Using Ensemble Models for Malicious Web Links Detection." In ICAART (3), 657-664. 2024.
50. Souri, Alireza, and Rahil Hosseini. "A State-of-the-Art Survey of Malware Detection Approaches Using Data Mining Techniques." Human-Centric Computing and Information Sciences 8, no. 1 (2018): 1-22.
51. Hess, S., P. Satam, S. Hariri, and G. Ditzler. "Malicious HTML File Prediction: A Detection and Classification Perspective." In ACS/IEEE International Conference on Computer Systems and Applications, 2018.
52. Soska, Kyle, and Nicolas Christin. "Automatically Detecting Vulnerable Websites Before They Turn Malicious." In 23rd USENIX Security Symposium (USENIX Security 14), 625-640. 2014.
53. Zhang, Wen, Yu-Xin Ding, Yan Tang, and Bin Zhao. "Malicious Web Page Detection Based on Online Learning Algorithm." In 2011 International Conference on Machine Learning and Cybernetics, vol. 4, 1914-1919. IEEE, 2011.
54. Rhode, Matilda, Pete Burnap, and Kevin Jones. "Early-Stage Malware Prediction Using Recurrent Neural Networks." Computers & Security 77 (2018): 578-594.
55. Abdelhamid, Neda, Fadi Thabtah, and Hussein Abdel-Jaber. "Phishing Detection: A Recent Intelligent Machine Learning Comparison Based on Models Content and Features." In 2017 IEEE International Conference on Intelligence and Security Informatics (ISI), 72-77. IEEE, 2017.
56. Ferreira, Marcelo. "Malicious URL Detection Using Machine Learning Algorithms." In Proceedings of the Digital Privacy Security Conference, 114-122, 2019.
57. Sheikhi, Saeid, and Panos Kostakos. "Safeguarding Cyberspace: Enhancing Malicious Website Detection with PSO-Optimized XGBoost and Firefly-Based Feature Selection." Computers & Security 142 (2024): 103885.
58. Singh, A. K. "Malicious and Benign Webpages Dataset." Data in Brief 32 (2020): 106304.
59. Manikanta, P., K. Nattar Kannan, and S. Padmakala. "Detection of Brain Stroke by Using the Novel Adaboost Classifier Algorithm Compared with Multi-Layer Perceptron." In 2024 IEEE Wireless Antenna and Microwave Symposium (WAMS), 1-6. IEEE, 2024.