OPTIMIZING MALICIOUS WEBSITE DETECTION THROUGH COMPARATIVE ANALYSIS OF MACHINE LEARNING TECHNIQUES

F. Malik[1*], A. U. Rahman[2], A. Ullah[3], R. Hussain[3], M. Javed[3] and S. Ullah[1]

[1]Department of Computer Science, Iqra National University Peshawar, Khyber Pakhtunkhwa (KPK), Pakistan, fazal.malik@inu.edu.pk, sanaullahk77@gmail.com;
[2]Faculty of Computer Information Science, Higher Colleges of Technology, Ras Al Khaimah Campus, United Arab Emirates, arahman@hct.ac.ae;
[3]Institute of Computer Science and Information Technology, University of Science and Technology Bannu, Khyber Pakhtunkhwa (KPK), Pakistan, ashrafbth@gmail.com, imrahmatwazir@gmail.com, drjaved@ustb.edu.pk;
[*] Corresponding Authors***:*** fazal.malik@inu.edu.pk***;***

ABSTRACT: The improvement of malware data exploitation risks, which appeared due to malicious websites, as well as an increase in their frequency, is results of modern threats. Modern methods for malicious website detection display a bad performance, producing multiple incorrect alarms, but fail to identify contemporary security threats correctly. More advanced malware website identification techniques are based on XGBoost systems combined with AdaBoost and Random Forest. The framework is composed of four phases: (1) Data Acquisition and Preliminary Analysis, utilizing a Kaggle dataset to discern key patterns; (2) Data Preprocessing and Model Implementation, which consists of data cleaning, normalization, and segmentation to train the model effectively; (3) Detection and Classification Evaluation, which computes performance metrics like precision, recall, F1-score, and accuracy; and (4) Comparative Analysis, where XGBoost outperforms traditional methods. The XGBoost model had a detection accuracy of 86.60% in its practice run since it generated less wrong outputs to show its capability in malware URL detection. Cybersecurity research needs machine learning in threat detection in order to eradicate human-based new threat evaluation processes and to demonstrate the need for sophisticated machine learning frameworks. The development of proven modern theoretical algorithms in malicious website detection should be researched upon because these algorithms show better effectiveness in research work.

Key words: Malicious websites, Cybersecurity, Performance Metrics, Phishing attacks, AdaBoost, Random Forest, XGBoost algorithms, Prediction.

(Received 05.07.2024 Accepted 01.09.2024)

INTRODUCTION

Legitimate websites are one of the most advanced and ongoing security threats against computer systems operating worldwide in the cyber environment. Such websites use deceptive methods to trick unsuspecting users into allowing their systems to experience different types of threats from breaches to system breakdowns. Strong detection systems must be developed to identify numerous cyber threats since they represent a crucial need [1]. The general conduct of rogue sites is installation of viruses and interference with running processes, all this in the process of fetching information through their fake download of files such as video codecs. A user is not safe from malware and spam as well as phishing. However, he has the latest security arrangement because a new detection system introduces itself as an automated threat blocking solution that analyzes the URL data through machine learning methodologies and eliminates all static content inspection mechanisms [2, 3].

They believe that machine learning approaches represent a new generation in security development in comparison with merely baseline signature-based and static rule-based detection practices. Random Forest along with the AdaBoost, and XGBoost serves to reduce both two accuracy indicators' values as well as latency that is necessary while detecting malicious website to save associated operational expenses at the operational costs. Even some of the newly developed modern detections systems suffer under the same constrained feature numbers with small datasets for feature extraction activities.

The researcher developed a machine learning system that combines six classifiers which involve Random Forest and Support Vector Machine and K-Nearest Neighbors and Multilayer Perceptron and Naive Bayes and Logistic Regression to detect phishing URLs through the analysis of components of the URL [4]. Detection of phishing URLs begins with prompt recognition of suspicious activity. The proposed research aims to identify unusual URL behaviors through supervised models including Random Forest (RF) and SVM. The proposal introduces a tracking system which gathers current attack campaign information instead of using traditional statistical data [5]. Machine learning (ML) integration represents an investigational response to modernize cybersecurity detection capabilities for rogue websites to enable preventative threat protection. The implementation of AI techniques especially XGBoost enhances detection speed and accuracy without involving human activity all the time, according to research [6]. Detecting phishing URLs quickly is still a priority since latest research applied supervised learning methods using Random Forest (RF) and Support Vector Machine (SVM) to detect anomalous URL activities. The technique monitors the ongoing attack activities to gather recent data which is not dependent on outdated datasets [7].

Malware, such as worms, spyware, viruses, Trojan horses, ransomware, and rootkits, enables unauthorized system access. Worms self-replicate, spyware collects user data, and viruses spread through files. Email viruses, Trojan horses, logic bombs, and key loggers exploit system vulnerabilities. Prediction in cybersecurity, similar to weather forecasting, uses past data to anticipate future events [8]. A variety of machine learning algorithms, including a hybrid Latent Semantic Decomposition (LSD) model, Decision Tree, Linear Regression, Random Forest, Naive Bayes, Gradient Boosting Classifier, K-Neighbors Classifier, and Support Vector Classifier, have been applied to enhance phishing detection accuracy [9].Additionally, models based on decision trees, random forests, support vector machines, and artificial neural networks were evaluated using the UCI phishing domains dataset[10]. To address limitations in traditional blacklist-based methods, a deep learning approach combining deep neural networks and variational autoencoders was developed for improved phishing detection amid evolving threats [11].

In previous studies, Random Forest Classifier (RFC) and Convolutional Neural Network (CNN) have been applied to evaluate multiclass malicious URL detection. Experimental findings indicate that the suggested features and techniques enhance the ability to detect risky URLs [12]. Another study employed neural networks, multiple Naive Bayes (NB) approaches, and Logistic Regression (LR) to classify websites as safe or risky, with Naive Bayes demonstrating superior performance. The methodology led to further development in order to identify between websites as secure or vulnerable with precision [13]. Many research studies have been done on various methods to detect dangerous URLs. An enormous dataset training set of 6000 samples was examined using big data RIPPER (Repeated Incremental Pruning to Produce Error Reduction) algorithm to identify bad URLs [14] . A dynamic security network developed using ML integrated SVMs with malicious URLs to prevent user-end attacks [15]. MalNet provides an advanced malware detection platform which combines opcode grouping with grayscale image processing to train both CNNs and LSTM systems [16]. Naïve Bayes proved successful at identifying dangerous URLs from large datasets through its analysis of lexical, network and content factors according to research [17]. SVMs and logistic regression, with their great power, could determine dangerous URLs based on site age and size of the URL that made it possible to protect the emergent online space [18].

Numerous AI-dependent feature computations such as decision trees, Random Forest (RF), Naive Bayes, Support Vector Machines (SVM), Neural Networks, and Instance-Based K (IBK) weak classifiers were used, and their accuracy was studied and compared [19]. Another study looked into resistance-based malicious URL detection that resolved the incompleteness and newly generated URLs through simpler algorithms with results put alongside SVM and Logistic Regression (LR) [20]. A machine learning approach for URL classification has been proposed to increase efficiency and accuracy using Random Forest for malicious URL detection [21]. Extensive analysis of malware detection tools using data mining techniques was also conducted along with detailed categorization of malware detection technologies and highlighting their essential components. Machine learning techniques such as SVM and RF were used for detection of malicious URLs, along with data reduction methods using instance selection to improve model performance [22]. Rapid growth of malicious websites threatens computer systems and causes malware diffusion while creating more severe security threats to users. Current research fails to provide the exact or efficient techniques for the prediction and categorization of threats, thus needing better techniques that can protect the users from malicious website threats.

The goal of this study is to overcome the accurate detection of malicious websites amidst the rapidly changing cyber threats. Traditional security techniques typically give out many false alarms and depict poor ability for new threat types. The system presented here suggests a four-stage malicious website prediction and classification scheme using Random Forest, AdaBoost, and XGBoost algorithms. The detection framework conducts data acquisition through a Kaggle dataset, and then data preprocessing occurs followed by RF, AdaBoost, and XGBoost algorithms for the detection of attacks and later evaluation with accuracy, precision, recall, and F1-score metrics. Through the joining of XGBoost classification capabilities with AdaBoost adaptability as well as RF robust properties the system achieved better accuracy rates to detect malicious websites more efficiently.

This study achieves three main contributions by lowering detection errors while improving the detection system's reliability through optimizing recall and precision and the adaptation of machine learning models to address modern cyber security threats. This process removes requiring human labor while simultaneously accelerating the speed needed for menace detection. The suggested approach delivers feasible security measures for systems alongside user defense solutions that promote international cybersecurity standards development.

According to the Literature Review section researchers present modern research techniques for analysis. The Methodology introduces both the designated framework together with its execution strategy. The section titled Results and Discussion presents an investigation of experimental results. The study ends with a concluding part that presents research recommendations alongside its summary.

LITERATURE REVIEW

Computer systems are most severely targeted by malicious web sites that are both spreading malware and introducing security vulnerabilities. In developing strategies for risk reduction in cyber security, researchers work with secure prediction methods along with classification models. Other works in the literature review study in addition to detection strategies, classification methods, along with prediction approaches for malignant web sites from different fields:

Approaches Utilizing Machine Learning for Detecting Malicious Websites: A study focused on improving detection methods by comparing ML algorithms such as Random Forest, Decision Tree, and K-Nearest Neighbor, while selecting the most relevant features [23]. Additionally, customer profiling through web browsing and email analysis was conducted to identify insider threats [24]. Various ML-based solutions were proposed to mitigate user risks, including those integrated within the Chrome browser [25].

Data-Centric and AI Techniques for Malware Detection: Several approaches have been used for effective malware detection through artificial intelligence (AI) and data mining. A framework was developed to improve report accuracy via a management rating system [26]. Both shallow and deep networks were employed to assess network efficiency in Windows executable files [27]. Real-time web spam detection was approached through link-based dispersion, analyzing both incoming and outgoing hyperlinks [28].

Deep Learning and Graph-Based Approaches: Deep learning techniques have also been employed to improve detection accuracy, with models such as Deep Graph Convolutional Neural Networks (DGCNNs) being trained on API call sequences and behavioral graphs. It was demonstrated that DGCNNs achieved performance comparable to Long Short-Term Memory (LSTM) networks in malware detection, attaining robust Area Under the Curve - Receiver Operating Characteristic (AUC-ROC) and F1-scores [29, 30]. Furthermore, web spam datasets were analyzed to evaluate the evolution of spam tactics, highlighting the necessity for effective filtering methods [31].

Innovative Spam and Phishing Detection Methods: The Consumer Internet of Things (CIoT) uses IoT technology to enhance daily ease. With the rapid growth of the Internet of Things, there is a significant surge in data from consumer devices. Web pages, as information carriers, expose CIoT systems to spam-related security threats. In response, page2vec, an intelligent feature extraction tool, and RFiRF, a unique classification algorithm, are proposed to detect web spam. Using a score propagation approach to determine goodness and badness scores via web graph links, Page2vec creates various web page properties [32]. Because spammers frequently alter the characteristics they use in their spam emails, conventional methods of email classification eventually lose their usefulness due to "Concept Drift." To solve this issue and ensure spam classification for eternity, a model is proposed [33]. Due to their constant internet connectivity, cell phones are vulnerable to phishing tactics, including smishing, which involves sending people phony SMS messages. In an ensemble learning approach, Random Forest and Support Vector Machine (SVM) models were employed, as well as feature extraction methods such as Term Frequency (TF) and Term Frequency-Inverse Document Frequency (TFIDF) [34].

Optimizing ML Models for Threat Detection: System infections have been predicted using data mining and artificial intelligence approaches, proving their capacity to identify new threat trends [35]. An enhanced network-based learning method was proposed for the identification of malicious web sites [36].

Advanced XGBoost-Based Models for Classification: A model for identifying various dangerous websites was developed using the Firefly algorithm for feature selection, followed by an improved XGBoost algorithm for classification. To achieve XGBoost optimization, the Particle Swarm Optimization (PSO) technique was applied [37]. This method contributed to stronger cybersecurity measures by demonstrating how well XGBoost performs malware detection and classification jobs.

Advanced machine learning techniques have been applied to COVID-19 pneumonia classification, sentiment analysis, accident prediction, and malicious website detection. For COVID-19, optimized Random Forest, XGBoost, CNNs, and AdaBoost used GitHub X-ray datasets and data augmentation to improve diagnostic accuracy [38-41]. In sentiment analysis, AdaBoost, XGBoost, and ANNs enhanced user review classification on the Google Play Store [42]. For accident prediction, Random Forest outperformed AdaBoost in analyzing dark data [43]. In cybersecurity, XGBoost and AdaBoost improved malicious URL detection using Kaggle datasets, reducing false positives/negatives. These studies demonstrate the effectiveness of advanced algorithms in enhancing decision-making and precision across domains [44, 45].

METHODOLOGY

As shown in Figure 1 and Algorithm 1, the proposed approach is divided into four stages. Data collection, tool selection, and preliminary analysis are all part of phase I. In Phase II, Random Forest, AdaBoost, and XGBoost models are used for data preprocessing, segmentation, and classification. Phase III focuses on evaluating the model using metrics from classification reports and confusion matrices. Phase IV evaluates results and contrasts them with previous studies to determine the possibility of the suggested solution in data science and machine learning.

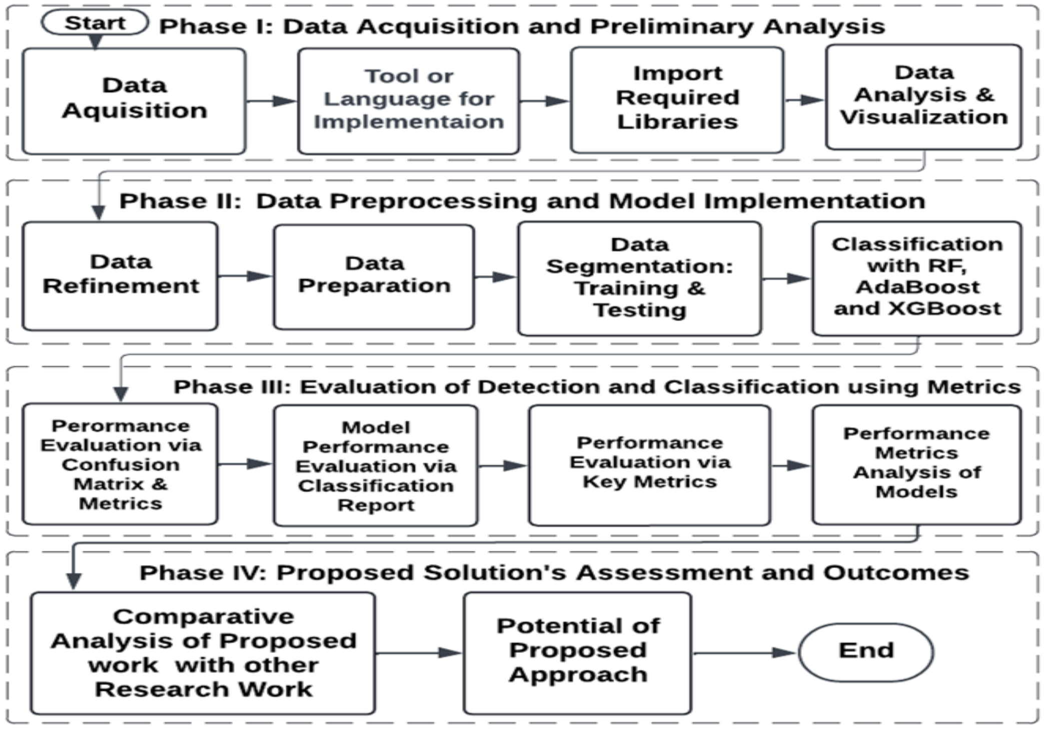
Start
Phase I: Data Acquisition and Preliminary Analysis
Data Aquisition
Tool or Language for Implementaion
Import Required Libraries
Data Analysis & Visualization
Phase II: Data Preprocessing and Model Implementation
Data Refinement
Data Preparation
Data Segmentation: Training & Testing
Classification with RF, AdaBoost and XGBoost
Phase III: Evaluation of Detection and Classification using Metrics
Perormance Evaluation via Confusion Matrix & Metrics
Model Performance Evaluation via Classification Report
Performance Evaluation via Key Metrics
Performance Metrics Analysis of Models
Phase IV: Proposed Solution's Assessment and Outcomes
Comparative Analysis of Proposed work with other Research Work
Potential of Proposed Approach
End

Figure 1. Block Diagram for Propose Solution.

```
Algorithm 01: Malicious Website Prediction Research
Input: Malicious Website Dataset from Kaggle
Output: Model Evaluation Results
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.   AdaboostClassifier ← InitializeAdaboostClassifier()
2.3.4.   RandomForestClassifier ← InitializeRandomForestClassifier()
2.3.5.   TrainModels(XGBoostClassifier, AdaboostClassifier, RandomForestClassifier, X, Y)
2.3.6.   optim_models ← OptimizeModels(XGBoostClassifier, AdaboostClassifier, RandomForestClassifier)
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)
```

First, ongoing research work is carefully examined, and harmful websites are found and eliminated utilizing up-to-date analyzers. Second, a straightforward method for predicting malicious endpoints is presented. Reducing the need for human involvement is the third objective. Lastly, results are evaluated and compared to previous studies, emphasizing accuracy within the suggested framework.

Phase I: Data Acquisition and Preliminary Analysis: Using supervised machine learning (ML), a predictive model identifies fraudulent websites using a Kaggle dataset with URL features like length and character composition. The classification techniques Random Forest together with AdaBoost and XGBoost function efficiently to identify malicious sites.

Dataset Acquisition: The proposed method utilized the "Malicious Webpages Dataset" which Singh and Kumar published on Kaggle during 2020 [46]. Web scraping extracted the data from MalCrawler during the period between November 2019 and March 2020. That global content database has become popular among machine learning practitioners for building and validating models that detect malicious websites. This dataset includes 1,781 instances separated through 21 distinctive features including URL length and special characters for malicious website separation. Our ML models consisting of Random Forest alongside AdaBoost and XGBoost executed their training and evaluation mechanisms using this dataset to boost detection precision.

Tool Selection for Implementation: A Python solution provides the best tool for implementation since it supports high-level functionality and extensive use in Machine Learning techniques through Random Forest and AdaBoost and XGBoost algorithms. Scientific researchers widely support this programming language which strengthens the decision to use Python. A lightweight Python web platform known as Jupyter Notebook will serve as the development environment to support efficient algorithm development through its specialized features.

Importing Essential Libraries for Dataset: Python welcomes Pandas together with NumPy and Matplotlib and Seaborn and Math libraries to enhance dataset manipulations and analysis during the implementation of Random Forest and the associated machine learning algorithms AdaBoost and XGBoost. Pandas functions as the primary tool for dataset reading and writing because it maintains extensive capabilities for CSV file management. The analytical capabilities of model training gain efficiency because NumPy enhances numerical and matrix data manipulation functions. Data visualization in this project is achieved through the use of Matplotlib and Seaborn libraries. The graphical representations including scatter plots are generated through the usage of these libraries for model performance evaluation. All necessary mathematical operations function through the Math library. These libraries offer complete features to manage numerical data with categorical data which are essential for implementing the proposed machine learning methods.

Exploratory Data Analysis and Visualization: A dataset import follows with extensive analytical visualization through Matplotlib and Seaborn libraries. Through graphical representations these libraries provide condensed information about dataset features for use in machine learning (ML) decisions. The baseline statistics for dataset attributes involve calculating mean, maximum and minimum values to serve as foundation data for Random Forest alongside AdaBoost and XGBoost frameworks

This dataset contains 1,781 instances that are divided into three distinct data types consisting of float64 for two columns and int64 for twelve columns along with object data type for seven columns as shown in Table 1 and Table 2. A mean value of 1,781 computes for malicious websites with a standard deviation measuring 27.5. A website can either have 16 or may extend to 249 as its minimum and maximum values.

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

Table 1. Statistical Description of Data.

	URL_ LENGTH	NUMBER_ SPECIAL_ CHARACTERS	CONTENT_ LENGTH	TCP_ CONVERSATION_ EXCHAGE	DIST_ REMOTE_ TCP_PORT	REMOTE_ IPS	APP_ BYTES
count	1781.000000	1781.000000	969.000000	1781.000000	1781.000000	1781.000000	1.781000
mean	56.961258	11.111.735	11726.927761	16.261089	5.472768	3.060640	2.982000
std	27.555586	4.549896	36391.809051	40.500975	21.807327	3.386975	5.605000
Min	16.000000	5.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	39.000000	8.000000	324.000000	0.000000	0.000000	0.000000	0.000000
50%	49.000000	10.000000	1853.000000	7.000000	0.000000	2.000000	6.720000
75%	68.000000	13.000000	11323.000000	22.000000	5.000000	5.000000	2.328000
max	249.000000	43.000000	649263.000000	1194.000000	708.000000	17.000000	2.362000

The dataset contains various attributes which demonstrate the syntactical features and lexical characteristics as well as network properties from URLs. All 21 features are used for training Random Forest, AdaBoost and XGBoost ML models and then these models are cross-compared. All attention in URL analysis goes toward features that closely mirror URL operations: URL_LENGTH, NUMBER_SPECIAL_CHARACTERS, and DNS_QUERY_TIMES. The incorporation of all features in the models allows them to detect subtle patterns that could have remained unnoticed in limited features which results in enhanced ability to distinguish harmful URLs from benign ones. The testing process aims to validate this overall feature-based strategy which demonstrates how well the model detects hazardous sites. Python alongside its libraries delivers an effective systematic analysis structure in cybersecurity which helps researchers develop Random Forest, AdaBoost and XGBoost models by enhancing their understanding of data. The platform establishes necessary foundations for ML models trials and assessment to unlock valuable insights from the malicious website information. The available tools enable systems to process the dataset using structured analysis and implement ML models for malicious website detection which reveals detailed information while supporting the implementation of ML methods for malicious URL detection.

Phase II: Data Preprocessing and Model Implementation

Data Refinement: Ensuring Data Consistency: The scikit-learn (sklearn) preprocessing libraries serve as tools to eliminate null and missing and irrelevant data values during data mining preprocessing. The current step produces error-free data through data refinement and generates an excellent base for analytical purposes. Data integrity receives protection from data wrangling procedures that detect and solve missing value problems. Table 3 demonstrates that server (1 value is absent), content_length (812 data points are missing) and dns_query_times (1 value is gone) from the specific features.

Table 2. 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

The replacement method in Table 4 uses mean imputation to prepare consistent data that will be analyzed by machine learning techniques like Random Forest, AdaBoost and Extreme Gradient Boosting (XGBoost).

Table 3. The process for replacing data that is 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

Data Preparation: The refined dataset becomes ready for analysis after data wrangling because it contains no irrelevant values and it stands prepared for training and testing. The proposed model requires the dataset as its primary decision-making component to apply machine learning techniques toward accurate malicious and non-malicious website identification using Random Forest, AdaBoost, and XGBoost.

Data Segmentation for Training and Testing: The dataset contains two distinct sets that separate dependent from independent variables where the target class variable y indicates malicious or non-malicious class yet X holds the status of predictor variables. The data segmentation process for training and testing utilizes X_train, X_test, y_train, and y_test elements from Scikit-learn model selection library techniques [47]. The established methodology creates a strong platform which supports the execution of Random Forest and AdaBoost along with XGBoost machine learning algorithms. The mathematical representation of the process states that we can define the dataset as:

$$D_{train} = \{(x_i, y_i)\}_{i=1}^{n_{train}} \qquad (1)$$

where $x_i \in \sim^m$ (independent variables) and $y_i \in \{0,1\}$ (dependent variable).

The model performance evaluation requires split methods from Scikit-learn model selection library to divide dataset ***D*** into training and testing sections. The mathematical definition for splitting this process describes it as follows:

$$D_{train} = \{(x_i, y_i)\}_{i=1}^{n_{train}} \quad (2)$$

$$D_{test} = \{(x_j, y_j)\}_{j=1}^{n_{test}} \quad (3)$$

where $n_{train} + n_{test} = n$ and typically $n_{train} = \alpha n$ and $n_{test} = (1-\alpha)n$, with α being a predefined proportion(e.g., α =0.8 for an 80-20 split).

The training set (X_{train}, y_{train}) and testing set (X_{test}, y_{test}) are defined as:

$$X_{train} = \{x_i \mid (x_i, y_i) \in D_{train}\},\ y_{train} = \{y_i \mid (x_i, y_i) \in l$$
…………………… *(4)*

$$X_{test} = \{x_j \mid (x_j, y_j) \in D_{test}\},\ y_{test} = \{y_j \mid (x_j, y_j) \in L$$
…………………… *(5)*

This approach establishes a robust foundation for the subsequent implementation of the proposed machine learning techniques, specifically Random Forest, AdaBoost, and XGBoost. It maintains the integrity of the model training and evaluation processes, facilitating the derivation of accurate performance metrics.

Classification Models Supervised learning serves as a fundamental approach in artificial intelligence (AI) and machine learning (ML), training models on labeled datasets to predict outputs for new data, particularly in classification and regression tasks. The model learns from known input-output pairs to assign labels to unseen data. This section utilizes AdaBoost (Adaptive Boosting), XGBoost (Extreme Gradient Boosting), and Random Forest techniques for the precise identification of malicious and non-malicious websites [48].

XGBoost is a very efficient ensemble model that builds multiple decision trees and iteratively corrects the errors. It is very efficient in feature-rich environments and very good at finding significant features for spam and fraud detection tasks. Another ensemble technique is Random Forest, which generates a collection of decision trees where each decision tree is trained on a random subset of the data and aggregates the predictions to improve classification accuracy and control overfitting. Together, these models exploit the strength of supervised learning to classify malicious and non-malicious websites correctly [49]. Let the dataset be represented as:

$$D = \{(x_i, y_i)\}_{i=1}^{N} \quad (6)$$

where $x_i \in \sim^d$ represents the feature vector related *i-th* instance, $y_i \in \{0,1\}$ is the label which is 0 for not malicious and 1 for the malicious one, and ***N*** represents the total number of instances. Every feature ***xj*** may be a representation of the length, special characters, and lexical characteristics of the URL.

- ***AdaBoost (Adaptive Boosting):*** The AdaBoost Classifier uses a training method that builds upon weak classification outputs by repeating the process while refining results from earlier models to lower prediction errors. The weighting scheme of misclassified instances becomes higher during the algorithm so predictive performance improves [50].

To address malicious sites the AdaBoost algorithm looks into signs of nefarious intent, assigning more evaluation to wrongly classified cases so precision increases. It focuses on tricky cases, exhibiting fewer indicators of fraudulence in an effort to detect subtle functional relations between constituents so that risk web sites could be identified successfully. AdaBoost performs very efficiently for cyber security as it works best in classifying malicious web sites from those of legitimate.

AdaBoost combines multiple weak classifiers $h_t(x)$ to form a strong classifier $H(x)$. The model is defined as:

$$H(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \quad (7)$$

where α_t is the weight assigned to each classifier based on its performance, calculated using the exponential loss function:

$$\alpha_t = \tfrac{1}{2}\ln\left(\tfrac{1-\epsilon_t}{\epsilon_t}\right) \quad (8)$$

with ϵ_t being the error rate of the $t-th$ classifier.

- ***XGBoost (Extreme Gradient Boosting)***

XGBoost (Extreme Gradient Boosting) is a robust supervised learning technique for large datasets, combining weak learners like decision trees to enhance predictive accuracy through boosting. It uses labeled data, adjusts errors iteratively, and identifies critical features, improving interpretability and performance in feature-rich datasets. XGBoost excels in classification tasks such as spam and fraud detection, leveraging regularization to prevent overfitting and supporting parallel processing for computational efficiency. The XGBClassifier in the xgboost library allows fine-tuning with parameters like colsample_bytree, learning_rate, max_depth, alpha, and n_estimators. In this research, XGBoost significantly boosts classification accuracy and speed, proving effective for malicious website detection, with detailed results in subsequent sections [51].

XGBoost implements gradient boosting, optimizing the following objective function:

$$L = \sum_{i=1}^{N} loss(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \qquad (9)$$

Where $\hat{y}_i$ is the predicted value, K is the number of trees, f_k is the $k-th$ tree, and $\Omega(f_k)$ represents a regularization term to control the complexity of the model:

$$\Omega(f) = \gamma T + \tfrac{1}{2}\lambda \,||\, \omega \,||^2 \qquad (10)$$

Here, T is the number of leaves in the tree, γ and λ are hyperparameters controlling tree complexity, and denotes the weights.

- ***Random Forest***

The Random Forest Classifier uses an ensemble approach to classify websites as malicious or non-malicious, leveraging Kaggle's "Malicious website URLs" dataset. By constructing multiple decision trees on randomly sampled subsets, it learns patterns from URL attributes like lexical and host-based features. Each tree votes on classifications, with the majority decision as the final output. This method enhances accuracy by combining diverse perspectives, reduces overfitting, and adapts well to imbalanced datasets, making it robust for distinguishing malicious from safe websites.

Random Forest builds an ensemble of decision trees using a subset of the training data and randomly selected features. For a dataset with N samples and M features, the Random Forest algorithm can be mathematically formulated as follows:

For each decision tree t, a bootstrap sample is drawn from the dataset. For each split in the tree, a random subset of m features (where $m < M$) is selected to determine the best split point. Each tree produces a prediction $h_t(x)$.

The final prediction $H(x)$ is the majority vote (for classification) or the average (for regression) of the individual tree predictions:

$$H(x) = \frac{1}{T}\sum_{t=1}^{T} h_t(x) \qquad (11)$$

Where T is the total number of decision trees in the Random Forest ensemble. Each tree contributes to the overall prediction, either by voting (in classification tasks) or by averaging (in regression tasks), x is the input sample for which a prediction is being made. This could represent a specific URL's features or other attributes in the context of malicious website detection.

RESULTS AND DISCUSSION

The Results and Discussion section assesses Random Forest, AdaBoost, and XGBoost techniques for detecting malicious websites, focusing on confusion matrix components (True Positives (TP), True Negatives (TN), False Positives (FP), False Negatives (FN)) and performance metrics (precision, recall, F1-score, accuracy). The experimental results confirm the model's performance capability, which indicates its potential applications in cybersecurity systems.

Phase III: Evaluation of Detection and Classification Using Metrics and Confusion Metrics

Performance Evaluation using confusion matrix metrics: A Confusion Matrix is used at this stage to evaluate the performance of ML classifiers Random Forest, AdaBoost, and XGBoost as they classify malicious and non-malicious websites. The actual and predicted labels appear in a Confusion Matrix that reveals TP, TN and both FP and FN elements. The calculated precision along with recall values enables determination of F1-score and accuracy to measure the overall performance of each classifier.

The Confusion Matrix enables the classification of classification errors and model performance evaluation. Detection of misidentified cases leads to efforts to enhance the model so that it maximizes the detection of malicious websites. Such an accurate method builds identification dependability for better cybersecurity applications in the field.

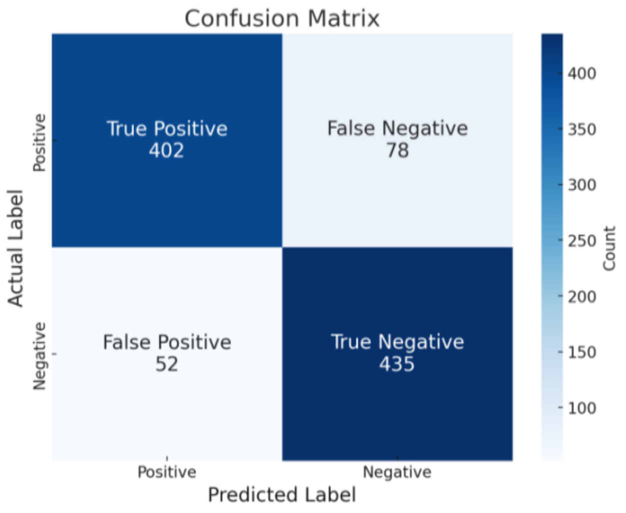

Figure 1. Confusion Matrix by XGBoost

For the XGBoost model, from the Confusion Matrix represented in Figure 2, it states that TP=402, which are well classified as malicious; TN=435, which are well classified as non-malicious; FP=52, which are non-malicious misclassified as malicious; FN=78, which are malicious misclassified as non-malicious. These values are very important for the computation of performance metrics, which will give insight into the accuracy of a classifier and its shortcomings

Figure 2 representing the Confusion Matrix in XGBoost The graphical visualization of the successful prediction rate for the predictor model is represented. The matrix contains actual and predicted labels, hence it indicates to which the model fails and opportunities for improvement can be done accordingly. That metric computed from the values of TP, TN, FP, and FN helps the cyber security practice by the appropriate identification of the malicious sites with the outcome result.

Performance Evaluation using key performance metrics: This section analyzes the malicious website detection models with regard to their accuracy, precision, recall, and F1 score. Average accuracy, computed from the confusion matrix, will determine how well it works against the actual results by comparing its predictions. Reliability is achieved through a classification report. Optimizing detection accuracy with minimum false positives continues to enhance the cybersecurity value of the model against the threats from online sources.

Performance is assessed using major parameters: accuracy, precision, recall, and F1 score.

Accuracy (AC) is defined as the number of correctly classified cases (both TP and TN) against the total number of cases, which represents the overall performance of the classification model, and is given by the following formula:

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

Precision (PR) is defined as the relevant results out of all the positive instances predicted. That is TP divided by TP+FP that is the sum of the correct malicious website number identified and total websites predicted as malicious. In this way, it measures accuracy in positive prediction regarding malicious websites:

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

Recall (RE) is the ratio of the number of relevant results to the total number of actual positive instances. It can be defined as TP/(TP + FN), where the ratio of TP to the sum of TP and FN gives the total number of malicious websites in the dataset that the model could correctly identify

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

F1 score is the harmonic mean of precision and recall; it is given by a single metric that has balanced precision with recall. It is helpful in cases with an uneven distribution of classes such as when there are far more non-malicious websites compared to malicious ones. The formula for calculating F1 Score is

$$F1-Score = 2\times\frac{PR\times RE}{PR+RE} \quad (4)$$

XGBoost depicts a great predictive performance that classifies malicious and genuine websites with great accuracy. Relevant metrics from the confusion matrix are applied for assessing the classification. Developed techniques such as augmentation and dimensionality reduction improve the accuracy of the prediction. Random Forest, AdaBoost, and XGBoost enhance the reliability of fraud detection. Above all, these models perform with appropriate sensitivity towards threats along with preservation of specificity, which amplifies their utility in cybersecurity and can add value to fraud prevention.

- ***XGBoost Performance Classification Report***

In this section, the XGBoost method—an effective technique of classification performance with respect to categorized structured data—has been followed. XGBoost is known as a technology of gradient boosting that enhances accuracy in prediction using boosting, decreasing error rates by boosting. Some results of XGBoost Model Performance Testing

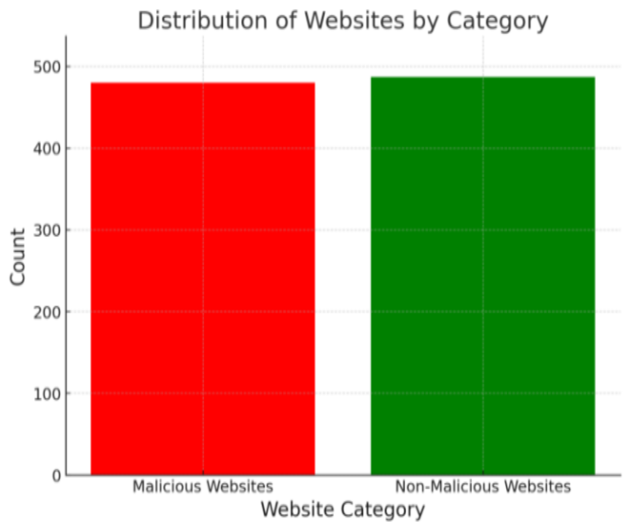

Figure 2. Report Classification Report Showing Distribution of Malicious and Non-Malicious Websites.

Figure 3 shows the performance metrics of the model-accuracy, precision, recall, and F1 score-obtained from confusion matrix analysis. The classification report shows that XGBoost performs well on a dataset of 967 websites with 487 non-malicious and 480 malicious. These metrics measure the accuracy and reliability of the model in cybersecurity applications.

Thus, the key performance metrics would involve a high true positive rate coupled with a minimal false positive. This is with the aim of reducing false alarms and detecting threat sites. There should be an adequate true negative rate so as to classify correct sites. Meanwhile, the FN rate is diminished to reduce possible undetected threats.

These metrics guide continuous optimization of XGBoost, balancing sensitivity (TP rate) and specificity (TN rate) to refine website classification, supporting effective cybersecurity defenses.

• ***Analysis of Performance Metrics for Proposed Models***

The performance of classifiers—Random Forest, AdaBoost, and XGBoost—is evaluated to better understand their ability to distinguish between malicious and benign websites. Table 1 summarizes key metrics such as Precision, Recall, F1 Score, and Accuracy for each model, providing an overall comparison of their strengths and weaknesses in the field of cybersecurity.

Performance Measure of classifiers.

Classifiers	Precision (%)	Recall (%)	F1 Score (%)	Accuracy (%)
Random Forest	77.30	74.10	75.70	75.00
AdaBoost	86.17	62.21	72.39	78.45
XGBoost	88.50	83.80	86.10	86.60

Out of a precision of 88.50%, XGBoost identified 88 out of 100 flagged websites as malicious and minimized the cases of false positives, which are very important in security. Random Forest has the least precision at 77.30%, while AdaBoost carries the highest at 86.17%. This means that it has more false positives. More false positives work against XGBoost. There are several ways to measure recall-a model's ability to recognize true dangers. With a recall of 83.80%, XGBoost outperforms Random Forest with 74.10% and AdaBoost with 62.21%. This means XGBoost will not miss any threats as it catches more hazardous websites. F1 Score balances precision and recall. The result of the use of XGBoost is reflected in the 86.10% F1 Score, demonstrating a good balance between true positive and false positives. Moderately, AdaBoost acts at F1 score of 72.39%. Random Forest is behind at 75.70% F1 score. Thus, the best overall balance provides XGBoost. Then, accuracy discloses the reliance on that model. XGBoost is reliable at 86.60%, while AdaBoost scores at 78.45%, and random forest at 75.00 %. These figures prove that XGBoost excels well beyond both competitors in terms of overall accuracy.

Despite both AdaBoost and XGBoost being good for malicious website detection, XGBoost has an edge over others in terms of precision, recall, F1 score, and accuracy, and hence it is the best model. It is strong enough to deal with tabular data and very well suitable for malicious URL detection application. Future work might be done to improve AdaBoost and XGBoost or hybrid models combining traditional machine learning methods with deep learning, further enriching the detection capabilities.

This work moves the frontiers of cybersecurity forward by evaluating predictive models for malicious website detection. It identifies XGBoost as the most effective classifier among traditional techniques. Table 1 indicates that XGBoost achieves a precision of 88.50%, recall of 83.80%, F1 score of 86.10%, and accuracy of 86.60%, which demonstrates its ability to distinguish between malicious and non-malicious websites.

The adoption of XGBoost demonstrates the commitment of the study to the improvement of classification techniques in cybersecurity. Performance with XGBoost was true and reliable to detect cyber threats with low risks of misclassification. In contrast, AdaBoost, which has 86.17% precision, its recall was just 62.21%, classifies only a few malicious websites. Random Forest with metrics of 77.30% precision, 74.10% recall, 75.70% F1 score, and 75.00% accuracy is balanced for applications requiring moderate accuracy and computational efficiency.

It further locks XGBoost as a useful algorithm and offers knowledge when more adaptive, accurate, and robust security defenses are built against ever-changing threats.

Phase IV: Assessment of the proposed solution and its outcomes: This section presents an analysis of the performance and robustness of our proposed models of machine learning in identifying a dangerous website using classification and prediction results. Comparison with earlier results allows us to carry out comparative analysis, shedding light on whether our technique has consistency and coherence.

Comparative Analysis of Proposed Work with Other Research Work: This section conducts an in-depth comparison of how well proposed machine learning models, such as Random Forest, AdaBoost, and XGBoost, perform compared with previous studies in terms of detecting malicious websites based on a Kaggle dataset, specifically the work by Al Tamimi and Saeed Ahmad [18] and the work by Malak Aljabri et al. [17]. The overall details for key performance metrics-Precision, Recall, F1 Score, and Accuracy-are discussed for each model across different studies in Table 2.

Comparative Analysis of Proposed Work with Other Research Work.

Authors	Data Set	Algorithm	Precision	Recall	F1-Score	Accuracy
Al Tamimi, Saeed Ahmad [18]	Malicious and Benign Websites Kaggle	Random Forest	68.00%	90.70%	77.70%	93.70%
Malak Aljabri, et al.[17]	Malicious and Benign Websites Kaggle	Random Forest	87.24%	100.00%	93.22%	95.15%
Our Propose Research work	Malicious and Benign Websites Kaggle	Random Forest	77.30%	74.10%	75.70%	75.00%
		AdaBoost	86.17%	62.21%	72.39%	78.45%
		XGBoost	88.50%	83.80%	86.10%	86.60%

Al Tamimi and Saeed Ahmad used Random Forest to classify malicious sites based on a Kaggle dataset with the precision of 68.00% and recall of 90.70%, F1-score of 77.70%, and accuracy of 93.70%. High recall reflects effective identification of the malicious websites, but a relatively lower precision represents a larger count of false positives. Though the accuracy is high, the model's performance on precision-recall curve imbalance cannot adequately explain the scenario.

Malak Aljabri et al. [17] applied Random Forest and achieved precision of 87.24%, recall of 100.00%, an F1-score of 93.22%, and accuracy of 95.15%. The 100% recall is impressive and indicates that the model correctly classifies all malicious websites. The precision was at a cost of reduced recall, indicating possible overfitting and too much sensitivity to malicious websites, which results in false positives. The high F1-score suggests a balanced trade-off between precision and recall.

Our comparative analysis includes three algorithms—Random Forest, AdaBoost, and XGBoost—revealing both similarities and differences. Our Random Forest implementation produced precision of 77.30%, recall of 74.10%, F1-score of 75.70%, and accuracy of 75.00%. Compared to previous studies, our Random Forest model shows lower precision and recall. Al Tamimi and Saeed Ahmad's model achieved a higher recall (90.70%), indicating superior detection of malicious websites, but at the cost of precision. Our model actually returns lower recall along with potentially fewer false positives.

The AdaBoost model in the study had 86.17% precision, 62.21% recall, 72.39% F1-score, and accuracy of 78.45%. Although precision was strong, there was a sacrifice in recall showing that AdaBoost fails to pick many malicious web sites. By contrast, the model by Aljabri achieves 100% recall but lags behind at precision. Here, our model using AdaBoost provides a balance but is poorly performing in regard to recall.

The XGBoost model performs the best, with precision of 88.50%, recall of 83.80%, F1-score of 86.10%, and accuracy of 86.60%. XGBoost strikes a balance between precision and recall, outperforming both Al Tamimi's and Aljabri's models. Its recall (83.80%) is higher than Al Tamimi's (90.70%), but its precision (88.50%) reduces false positives. XGBoost also outperforms Random Forest and AdaBoost in accuracy, indicating superior reliability and effectiveness.

- ***Key insights:*** Key insights include the trade-off between precision and recall, with XGBoost offering the best balance. The F1-score is critical when precision and recall must be balanced. Our XGBoost model, with an F1-score of 86.10%, is ideal for cybersecurity applications. Accuracy, while important, can be misleading in imbalanced datasets, as seen in Al Tamimi's model.

XGBoost provides the best balanced and reliable solution for detecting malicious websites and offers the overall best performance. Future research might integrate traditional models with advanced techniques like deep learning to enhance the detection capabilities..

Conclusion: This study also demonstrates that malicious websites continually pose a security threat, attacking the user even with the presence of security defenses by way of malware. It calls for enhancing detection systems through URL data and machine learning techniques for proactive blocking, with no content inspection. The study enhances malicious URL detection through Random Forest, AdaBoost, and XGBoost algorithms, with XGBoost outperforming others in key performance metrics: 86.60% accuracy, 88.50% precision, 83.80% recall, and 86.10% F1-score. These results underscore XGBoost's ability to accurately distinguish between malicious and benign websites, surpassing previous methods in precision and recall. A comparative analysis shows the superiority of XGBoost over Al Tamimi's and Aljabri's models. While Al Tamimi's model achieves higher recall (90.70%), XGBoost's higher precision (88.50%) reduces false positives, providing better overall performance. Additionally, XGBoost outperforms both Random Forest and AdaBoost in accuracy, demonstrating its reliability and effectiveness. This is a critical finding, which states that XGBoost provides the best balance between recall and precision. The model of XGBoost is ideal for cybersecurity applications due to having a required F1-score of 86.10%. Al Tamimi's model did demonstrate that accuracy is paramount, yet in unbalanced datasets, it is misleading. In this study, the impact that both algorithm and feature choice have on accuracy in detection would clearly give insightful value about the development in detection systems. Scalability and adaptability in machine learning algorithms indicate good prospects for being integrated into current cybersecurity frameworks.

Future research would then focus on in-depth feature selection, fusion approaches, and designing measures that are adaptive for real-time threat detection. To tackle the constantly evolving cyberthreats, there would be a need for an effective evaluation through different datasets and novel models.

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), (2023), 12(2), pp. 1704–1712. doi:10.21275/es24101104007.
2. Naim, O., Cohen, D., and Gal, I. B. "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.
3. 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.
4. 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.
5. 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.
6. Kaur, Ramanpreet, Dušan Gabrijelčič, and Tomaž Klobučar. "Artificial intelligence for cybersecurity: Literature review and future research directions." Information Fusion 97 (2023): p-101804.
7. 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.
8. 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.
9. Karim, Abdul, Mobeen Shahroz, Khabib Mustofa, Samir Brahim Belhaouari, and S. Ramana Kumar Joga. "Phishing detection system through hybrid machine learning based on URL." in IEEE Access, vol. 11, pp. 36805-36822, 2023, doi: 10.1109/ACCESS.2023.3252366.
10. Alnemari, Shouq, and Majid Alshammari. "Detecting phishing domains using machine learning." Applied Sciences 13, no. 8 (2023): 4649.
11. 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.
12. Mumu, Mahmuda Haque, and Tanzina Aishy. "Malicious URL detection using machine learning and deep learning algorithms." PhD diss., East West University, 2023.
13. 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. doi: 10.35377/saucis...1273536.
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), pp. 1296-1301. IEEE, 2017.
15. Jeenal, R., Preethi, G., Praveena, A., and Preethi, A. "Malicious URL Detection Using Machine Learning Techniques." International Journal of Computer Applications 8, no. 3 (2019): 1-5.
16. Yan, Jinpei, Yong Qi, and Qifan Rao. "Detecting malware with an ensemble method based on deep neural network." Security and Communication Networks 2018, no. 1 (2018): 7247095.
17. 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, no. 1 (2022): 3241216.

18. Al Tamimi, Saeed Ahmad. "Detecting malicious websites using machine learning." Journal of Cyber Security Technology, vol. 4, no. 1, 2020, pp. 45-60.
19. Islam, Mazharul, and Nihad Karim Chowdhury. "Phishing websites detection using machine learning based classification techniques." In International Conference on Advanced Information and Communication Technology, Chittagong, Bangladesh, vol. 10, no. 9, pp. 4393-4402. 2016.
20. Abdi, Farhan Douksieh, and Lian Wenjuan. "Malicious URL detection using convolutional neural network." Journal International Journal of Computer Science, Engineering and Information Technology 7, no. 6 (2017): 1-8.
21. 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.
22. Abad, Shayan, Hassan Gholamy, and Mohammad Aslani. "Classification of malicious URLs using machine learning." Sensors 23, no. 18 (2023): 7760.
23. COSTE, Claudia-Ioana. "MALICIOUS WEB LINKS DETECTION-A COMPARATIVE ANALYSIS OF MACHINE LEARNING ALGORITHMS." Studia Universitatis Babeş-Bolyai Informatica (2023): 21-36.
24. 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), pp. 1-6. IEEE, 2018.
25. Desai, Anand, Janvi Jatakia, Rohit Naik, and Nataasha Raul. "Malicious web content detection using machine leaning." In 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), pp. 1432-1436. IEEE, 2017.
26. Du, J., Jiang, C., Zhang, H., Ren, Y., & Poor, H. V. (2018). Peer Prediction Based Trustworthiness Evaluation and Trustworthy Service Rating in Social Networks. IEEE. doi:10.1109/TNSE.2018.2813044.
27. 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.
28. Egele, Manuel, Clemens Kolbitsch, and Christian Platzer. "Removing web spam links from search engine results." Journal in Computer Virology 7 (2011): 51-62.
29. 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.
30. de Oliveira, Angelo Schranko, and Renato José Sassi. "Behavioral malware detection using deep graph convolutional neural networks." Authorea Preprints (2023).
31. 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), pp. 40-49. IEEE, 2012.
32. 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).
33. Mohammad, Rami Mustafa A. "A lifelong spam emails classification model." Applied Computing and Informatics 20, no. 1/2 (2024): 35-54.
34. Xu, Hongsheng, Akeel Qadir, and Saima Sadiq. "Enhancing Mobile Cybersecurity: Smishing Detection Using Ensemble Learning and Smote." Available at SSRN 4875342. 2024.
35. 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.
36. Zhang, Wen, Yu-Xin Ding, Yan Tang, and Bin Zhao. "Malicious web page detection based on on-line learning algorithm." In 2011 International Conference on Machine Learning and Cybernetics, vol. 4, pp. 1914-1919. IEEE, 2011.
37. Sheikhi, Saeid, and Panos Kostakos. "Safeguarding cyberspace: Enhancing malicious website detection with PSOoptimized XGBoost and firefly-based feature selection." Computers & Security 142 (2024): 103885.

38. Malik, Fazal, Muhammad Suliman, Shehla Shaha, Muhammad Qasim Khan, and Abd Ur Rub. "Optimizing Pneumonia Diagnosis during COVID-19: Utilizing Random Forest for Accurate Classification and Effective Public Health Interventions." Journal of Computing & Biomedical Informatics 7, no. 01 (2024): 297-312.
39. Malik, Fazal, Muhammad Suliman, Muhammad Qasim Khan, Noor Rahman, and Mohammad Khan. "Optimized XGBoost-based model for accurate detection and classification of COVID-19 pneumonia." Journal of Computing & Biomedical Informatics 7, no. 02 (2024).
40. Suliman, Muhammad, Fazal Malik, Muhammad Qasim Khan, Ashraf Ullah, Noor Rahman, and Said Khalid Shah. "A Convolutional Neural Network (CNN) Based Framework for Enhanced Diagnosis and Classification of COVID-19 Pneumonia." VAWKUM Transactions on Computer Sciences 12, no. 2 (2024): 220-240.
41. Suliman, Muhammad, Fazal Malik, Muhammad Qasim Khan, Irfan Ullah, and Abd Ur Rub. "Integrating data augmentation with AdaBoost for effective COVID-19 pneumonia classification." Journal of Computing & Biomedical Informatics 7, no. 01 (2024): 590-605.
42. Khan, Muhammad Qasim, Fazal Malik, and Noor Rahman. "Optimized Sentiment Classification of Google Play Store App Ratings Using Advanced Machine Learning Models." VFAST Transactions on Software Engineering 12, no. 4 (2024): 252-266.
43. Shah, Masroor, Fazal Malik, Muhammad Suliman, Noor Rahman, Irfan Ullah, Sana Ullah, Romaan Khan, and Salman Alam. "Dark Data in Accident Prediction: Using AdaBoost and Random Forest for Improved Accuracy." Journal of Computing & Biomedical Informatics 7, no. 02 (2024).
44. Malik, Fazal, Muhammad Suliman, Muhammad Qasim Khan, Noor Rahman, Khairullah Khan, and Muhammad Khan. "Optimizing malicious website detection with the XGBoost machine learning approach." Journal of Computing & Biomedical Informatics 7, no. 02 (2024).
45. Malik, Fazal Malik Fazal, Muhammad Suliman Suliman, Irfan ullah Irfan, Shehla Shah Shehla, and Asiya Bibi Asiya. "Enhancing Cyber Security: A Holistic Strategy for Advanced Malicious Website Prediction Using AdaBoost Algorithm." Lahore Garrison University Research Journal of Computer Science and Information Technology 8, no. 3 (2024).
46. Singh, A. K. "Malicious and benign webpages dataset." Data in brief 32 (2020): 106304.
47. Manikanta, P., K. Nattar Kannan, and S. Padmakala. "Detection of Brain Stroke by using the Novel Adaboost Classifier Algorithm Classifier Algorithm Compared with Multi-layer Perceptron." In 2024 IEEE Wireless Antenna and Microwave Symposium (WAMS), pp. 1-6. IEEE, 2024.
48. Wynants, Laure, Ben Van Calster, Marc MJ Bonten, Gary S. Collins, Thomas PA Debray, Maarten De Vos, Maria C. Haller et al. "Systematic review and critical appraisal of prediction models for diagnosis and prognosis of COVID-19 infection." MedRxiv (2020): 2020-03.
49. Soni, Kartik M., Amisha Gupta, and Tarun Jain. "Supervised machine learning approaches for breast cancer classification and a high performance recurrent neural network." In 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), pp. 1-7. IEEE, 2021.
50. de Giorgio, Andrea, Gabriele Cola, and Lihui Wang. "Systematic review of class imbalance problems in manufacturing." Journal of Manufacturing Systems 71 (2023): 620-644.
51. Simplilearn, "What is XGBoost Algorithm in Machine Learning?" Simplilearn, 2023. Available Online: https://www.simplilearn.com/what-is-xgboost-algorithm-in-machine-learning-article. [Accessed on : 26-08-2024].