Journal of Infrastructure, Policy and Development 2024, 8(15), 8848.
https://doi.org/10.24294/jipd8848

Article

Identifying suspicious internet threat exchanges using machine learning algorithms to ensure privacy and cybersecurity in the USA

Syeda Farjana Farabi[1], Barna Biswas[2], Md Imran Sarkar[2], Nur Mohammad[2], Mohammad Zahidul Alam[2], Rakibul Hasan[1,*], Masuk Abdullah[3]

[1] Department of Business Administration, Westcliff University, 17877 Von Karman Ave., Irvine, CA 92614, United States
[2] Department of Technology & Engineering, Westcliff University, 17877 Von Karman Ave., Irvine, CA 92614, United States
[3] Faculty of Engineering, University of Debrecen, Ótemető strt. 2-4, 4028 Debrecen, Hungary
*** Corresponding author:** Rakibul Hasan, r.hasan.179@westcliff.edu

CITATION

Farabi SF, Biswas B, Sarkar MI, et al. (2024). Identifying suspicious internet threat exchanges using machine learning algorithms to ensure privacy and cybersecurity in the USA. Journal of Infrastructure, Policy and Development. 8(15): 8848.
https://doi.org/10.24294/jipd8848

ARTICLE INFO

Received: 28 August 2024
Accepted: 23 September 2024
Available online: 13 December 2024

Abstract: The usage of cybersecurity is growing steadily because it is beneficial to us. When people use cybersecurity, they can easily protect their valuable data. Today, everyone is connected through the internet. It's much easier for a thief to connect important data through cyber-attacks. Everyone needs cybersecurity to protect their precious personal data and sustainable infrastructure development in data science. However, systems protecting our data using the existing cybersecurity systems is difficult. There are different types of cybersecurity threats. It can be phishing, malware, ransomware, and so on. To prevent these attacks, people need advanced cybersecurity systems. Many software helps to prevent cyber-attacks. However, these are not able to early detect suspicious internet threat exchanges. This research used machine learning models in cybersecurity to enhance threat detection. Reducing cyberattacks internet and enhancing data protection; this system makes it possible to browse anywhere through the internet securely. The Kaggle dataset was collected to build technology to detect untrustworthy online threat exchanges early. To obtain better results and accuracy, a few pre-processing approaches were applied. Feature engineering is applied to the dataset to improve the quality of data. Ultimately, the random forest, gradient boosting, XGBoost, and Light GBM were used to achieve our goal. Random forest obtained 96% accuracy, which is the best and helpful to get a good outcome for the social development in the cybersecurity system.

Keywords: cybersecurity; social development; random forest; gradient boosting; XGBoost; machine learning

1. Introduction

People are interested in working online through the internet. They are using the internet for different prospects. Some it for their office work, and some for their enjoyment. Emerging Internet electronic goods are distinguished by enough computation to accomplish the purpose in a tiny, frequently portable form factor, in contrast to PCs, which over the past 20 years have been characterized by ever-increasing calculation capabilities (Buss, 2002). Similarly, the cyber-attacks are increasing day by day. A lot of people use different websites and unauthorized software daily. It's likely that everyone visiting some websites or using some software that can give access to one's personal or valuable data to a thief. However, sources of systemic risk include viruses, hackers, and software authorized software usage. Many businesses that are in the same situation (Chowdhury and Arefeen, 2011). In the meantime, hackers or thieves are advantage of this to collect important company data. They can use different techniques to take advantage. Phishing, SQL injection, viruses,

XSS, ransomware, worms, malware, and DDoS attacks are the most frequent dangers to web security (Fortinet, 2024). Many technologies have been invented to detect unauthorized web links. Phishing URL detection is common. A technology built using machine learning with wrapper feature selection to detect phishing website URLs (Swathi et al., 2023). It worked well compared to other existing models. But some improvements are needed to enhance the detection ability. During the coronavirus pandemic, Malevolent assailants have viewed this as a chance to carry out assaults to further their nefarious objectives and earn money. Ransomware attacks compromise healthcare systems and jeopardizing resources, including patient data confidentiality and integrity. Phishing assaults are exploiting people with content relating to COVID-19 (Khan et al., 2023). According to the majority of reports, since the pandemic's beginning, there has been a great increase in fraud and malware attacks (Gallagher and Brandt, 2020). Unauthorized websites and software are available everywhere. People are not able to identify the authored website and software. Many authorized software companies provide protected and safe software. It is very easy to identify the safe software. Everyone trusts many companies like Google, Microsoft, Apple, and so on. It is very difficult to locate safe and protected websites if we come to websites. Hackers may be able to access the network through a website that has security flaws and vulnerabilities. Many of the techniques don't require any prior hacking knowledge. These methods can be used with minimal understanding of these attacks (Johora et al., 2024; Kamruzzaman et al., 2024; Linkon et al., 2024). If a website has any susceptible code, any user can carry out these attacks (Jamil et al., 2018). It is essential to identify a safe website. Some methods will help detect authorized websites. Over the decades, numerous effective solutions have been implemented to combat phishing attacks. Still, no one approach fits all situations, as attack strategies are ever-evolving (Tang and Mahmoud, 2021). So, it is tough to identify whether it is safe or unsafe. That's why it is crucial to have a system that can detect with a high accuracy rate whether the website is protected.

Public safety, as well as economic and national security, are threatened by cybercrime. The Federal Bureau of Investigation (FBI) is the primary federal organization investigating cyberattacks and intrusions (Federal Bureau of Investigation, 2023). As a result, the number of cyberattacks in the USA is increasing daily. Cyberattacks can be avoided if early detection is possible before accessing a website. But, without robust automated technologies, identifying unauthorized websites is difficult. An artificial neural network that performs well on complex tasks must be carefully constructed by carefully selecting and extracting features (Akter et al., 2024; Bhuyan et al., 2024; Biswas et al., 2024). It might be able to quickly identify any unauthorized website by using machine learning algorithms. Early cyberattack detection is essential for lowering hacking. With machine learning models and historical cybersecurity data, the system becomes more accurate (Habibur Rahman Sobuz et al., 2024; Hossain et al., 2024; Mohammad et al., 2024; Nilima et al., 2024; Sobuz, Al, et al., 2024). This is the reason this method has evolved to detect these kinds of hacking at an early stage with speed. This paper introduces a system using machine learning with a high accuracy rate that can detect unsafe website links as fast as possible. A unique feature extraction was added to the dataset to enhance the quality of data. A few models with various algorithms were utilized to increase accuracy.

2. Literature survey

The frequency of cyberattacks is rising daily. Many people use unapproved software and other websites basis daily. Everyone using software or accessing certain websites probably has access to personal information that a thief could steal. The safe website must be identified immediately. A few techniques will be available to assist in identifying the approved website. It might be challenging to determine whether something is safe, though, at times. Therefore, having a system that can determine whether a website is protected is vital. Identifying hazardous websites is expensive and time-consuming since specialized methods and tools are required (Görgün, 2022; Johora et al., 2021; Manik, 2022). To lessen this issue, more training and experience are also required. This is why machine learning has been used recently to investigate automatic unprotected website detection. In the following lines, some current research on the automatic detection of cyberattacks is discussed.

Shetty et al. (2023) developed a technique for detecting malicious URLs through machine learning. The 651,191 URLs in the dataset were gathered via Kaggle. The four different categories of URLs in the dataset. Before any models were utilized, data preprocessing was carried out, including data reduction, transformation, cleansing, and feature engineering. They also separated the dataset into test and train sets. They have run the dataset through three different models to improve the results. The author used XGBoost, LightGBM, and random forest to get the best accuracy. The random forest fared well in terms of F1 score for each class. Meanwhile, LightGBM achieved the highest F-1 score for the single innocuous class.

An ensemble machine learning phishing attack detection system was developed by Innab et al. (2024). Two phishing datasets were used. There are 11,055 occurrences with 32 features in the first dataset. Ten thousand examples with fifty features make up the second dataset. There are two groups in both datasets: legitimate and phishing. Before any models were utilized, the data underwent preprocessing, which included cleaning, normalization, transformation, and reduction. They also separated the dataset into test and train sets. They separately applied seven models to each of the two datasets to improve the results. With 97% accuracy in the first dataset, the XGB did well. However, all models improved to about 100% accuracy.

A contemporary machine-learning phishing assault detection method was developed by Mosa et al. (2023). The Kaggle dataset, which has over 11,000 URLs with 30 characteristics, was gathered. Before any models were utilized, data preprocessing was carried out, including cleaning, feature extraction, transformation, and reduction. They also separated the dataset into test and train sets. The author used NB, NN, and AdaBoost to achieve the best accuracy possible. With an accuracy of 95.43%, AdaBoost achieved the highest accuracy.

Mahdi Bahaghighat et al. developed a machine-learning approach for phishing assault identification that is very accurate (Bahaghighat et al., 2023). The Phishing Websites Dataset was collected and consists of 88,647 URLs with 111 characteristics. There was an unbalanced ratio between phishing and non-phishing. Data preprocessing, including feature selection, Synthetic Minority Oversampling-Edited Nearest Neighbor (SMOTE-ENN), and constant features reduction, was done before any models were used. Additionally, they divided the dataset into train and test sets.

The author employed random forest, support vector machine (SVM), XGBoost, k-nearest neighbors (KNN), Naive Bayes, and logistic regression to achieve the highest accuracy (Mottakin et al., 2024; Sobuz, Aditto, et al., 2024; Sobuz, Jabin, et al., 2024). The maximum accuracy was reached by XGBoost, with a score of 99.22%.

Using machine learning, Shahriar et al. (2020) created a case-based cybersecurity detection system to stop fraud. A Kaggle dataset was utilized in this experiment. The train and test sets of the dataset were separated. Data preprocessing, which included data cleaning, integration, transformation, and reduction, was done before any models were used. Docker operates over the entire system. The author reduced the unbalanced data using a variety of strategies. The author employed logistic regression to get the maximum accuracy possible. Last but not least, logistic regression offers a maximum accuracy of 99%.

A machine learning-based classification technique for malware family classification was presented by Chen et al. (2020). In this paper, the authors proposed a method for detecting malware family classification based on autonomous machine learning. The authors used the dataset they collected from Kaggle (Microsoft Malware Classification Challenge) to ascertain the outcomes. Preprocessing was done on the data before it was posted to the system. However, the researchers used SVM with adaptive load balancing algorithm (ALBL) to build this system. Finally, the scientists found that the SVM with ALBL had better outcomes compared to SVM.

Hassan et al. (2019) developed a technique for identifying network intrusion using machine learning in an effort to enhance availability. This experiment used a publicly available dataset named ES. ES is a NoSQL database. There is no missing value in this dataset. Prior to using any models, the data was preprocessed. The author employed various methodologies for tasks such as data mining and cleaning. For the best accuracy, the author used unsupervised learning for intrusion detection. ELK stack was used in the unsupervised learning to increase the outcome ability.

Aksu and Aydin (2018) created a machine learning and deep learning-based port scan attempts detection system. A CICIDS2017 dataset from the Canadian Institute with 286,467 records and 85 features was used in this experiment. Data preprocessing, including data cleaning, integration, transformation, and reduction, was done before any models were used. The author lessened the unbalanced data by using resampling procedures. Additionally, they divided the dataset into train and test sets and applied binary classification to the desired feature. The training set was split into two IDS models. One was for SVM, and the other one was for deep learning. The author employed logistic regression, SVM, and deep learning to achieve maximum accuracy. The lowest accuracy of 69% was attained with SVM. Last but not least, deep learning offers a maximum accuracy of 97% (Datta, Sarkar, et al., 2024; Habibur Rahman Sobuz et al., 2024; Hasan et al., 2023).

Z. S. Lee et al. (2020) introduced a machine learning-based classification method for anomalies in a rail supervisory control and data acquisition (SCADA) detection. The authors of this research presented an autonomous machine learning-based approach for detecting anomalies in a rail SCADA. To determine the results, the authors used the network traffic data dataset gathered from a legal source. The dataset underwent different methods to eliminate oversampling. The data underwent preprocessing before being uploaded to the system. However, the researchers built the

system using KNN, LinearSVC, Random Forest, and Gaussian Bayes. Ultimately, the researchers discovered that the KNN classifier outperformed the others with an accuracy of 100% (Mehedi et al., 2024).

Zhao et al. (2020) introduced a classification technique based on machine learning for Transmission Control Protocol (TCP) security action detection. The researchers in this study proposed an autonomous machine learning-based method for identifying TCP security action. The authors used the dataset they had collected from UCI to ascertain the outcomes. The dataset was processed using a hybrid technique to remove both oversampling and undersampling. Before being submitted to the system, the data was preprocessed. The researchers built the system using a neural network, SVM, AdaBoost, and logistic regression. Finally, with an accuracy of 98%, the researchers found that the AdaBoost performed better than the others.

Lee et al. (2020) introduced a machine learning-based classification method for encrypted malware traffic detection. The authors determined the results using the dataset they had gathered from a legal source, which had 31 flow features. To balance the dataset, different techniques were applied. Preprocessing was done before uploading the data to the system. However, the researchers employed SVM with Stochastic Gradient Descent (SGD), Passive Aggressive, and Gaussian Naive Bayes to construct this system. Ultimately, the researchers discovered that the SVM with SGD had done well and had a higher accuracy of 94% (Datta, Islam, et al., 2024; M. M. H. Khan et al., 2023; Sobuz, Joy, et al., 2024).

This work will aid in preventing cyberattacks by detecting dangerous website URLs through the use of machine-learning techniques. From the Kaggle dataset, 651,191 URLs in total were extracted. Five distinct sources provided the information for this dataset. Feature selection yields the important features. Unnecessary features were eliminated. In order to train our models, we used extensive data preprocessing, data cleaning, and well-known classifiers. The dataset was trained using random forest, gradient boosting, XGBoost, and Light GBM, which simplified the process of obtaining results of a high caliber. We used several preparation techniques to ensure that our dataset was noise-free. In summary, the accuracy of the random forest model outperformed the other models.

In this work, dangerous website URLs detection is achieved by machine learning. The following are some noteworthy contributions made by this work:

- A significant contribution of this study is the application of pre-processing techniques to the collected data, which consists of 651,191 URLs.
- Random forest, gradient boosting, XGBoost, and Light Gradient Boosting Machine (GBM) were used to identify the best outcomes and applied to the dataset to categorize dangerous website URL detection.
- The use of random forest, gradient boosting, XGBoost, and Light GBM machine learning to gain a good result makes this work different from others. Language models were implemented to create an automated dangerous website URL detection that utilizes the Kaggle dataset to decrease cyber-attacks, which is what makes this unique research compared to others.

3. Proposed methodology

Our main goal is to create a system to detect unsafe URLs to extend cybersecurity. At first, the dataset should be polished and prepared for machine learning models. After that, we can get the best outcome using different approaches and machine learning models. Machine learning models, preprocessing, datasets, and other topics have all been briefly covered in this section.

3.1. Dataset

The dataset was collected using Kaggle, consisting of 651,191 URLs (Siddhartha, 2024). Five distinct sources provided the information for this dataset. Two columns make up this dataset: one is for URLs, and the other is for URL kinds. The dataset could not provide many important details due to significant confidentiality constraints. The number of safe URLs is around 428,103, phishing links are 94,111, defacement URLs are 96,457, and malware URLs are 32,520. This dataset offers four different kinds of URLs. **Figure 1** is a picture of a sample of the dataset.

	url	type
0	br-icloud.com.br	phishing
1	mp3raid.com/music/krizz_kaliko.html	benign
2	bopsecrets.org/rexroth/cr/1.htm	benign
3	http://www.garage-pirenne.be/index.php?option=...	defacement
4	http://adventure-nicaragua.net/index.php?optio...	defacement
5	http://buzzfil.net/m/show-art/ils-etaient-loin...	benign
6	espn.go.com/nba/player/_/id/3457/brandon-rush	benign
7	yourbittorrent.com/?q=anthony-hamilton-soulife	benign
8	http://www.pashminaonline.com/pure-pashminas	defacement
9	allmusic.com/album/crazy-from-the-heat-r16990	benign

Figure 1. Sample dataset.

3.2. Pre-processing

Data preparation is the process of producing unprocessed data for machine learning algorithms. This is the first step in building a machine-learning model. The modifications we apply to our dataset before submitting it to the software are called "pre-processing." Data preparation is one way to convert the unprocessed data into an error-free data set. In this work, many pre-processing methods have been used.

EDA (exploratory data analysis): In data analysis, it is significant. When data is visualized using EDA, choosing the right processing method for a raw dataset becomes noise-free and easier. The "type" target feature was examined. This column has four different kinds of URLs. The details of this 'type' column have been discovered following the distribution of URLs for each category. 32,520 are harmful URLs, 94,111 are phishing URLs, 96,457 are defacement URLs, and 428,103 are safe

URLs. There were 651,191 total URLs in the dataset. 65.7% are safe URLs, and 34.3% of URLs are unsafe. **Figure 2** shows the distribution of URLs for each category.

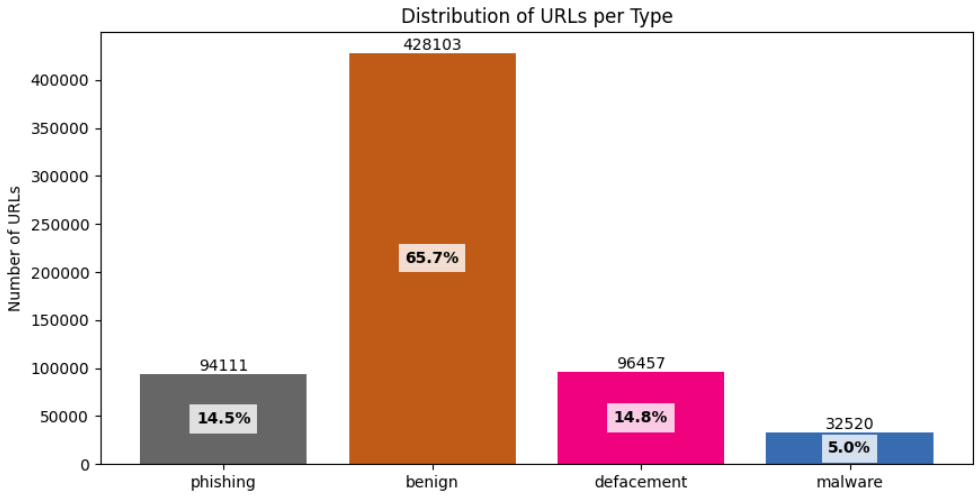

Figure 2. Class feature visualization.

Url_len column added to the dataset, where the length of each URL is stored. The defacement URLs are long compared to others. Safe and malware links are similar. The phishing URLs are short. **Figure 3** represents the URL length per category.

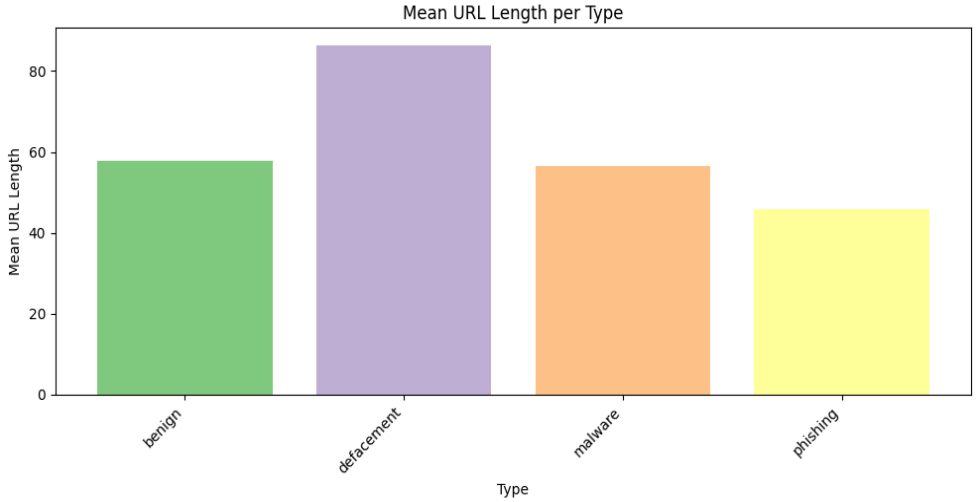

Figure 3. Means of URLs length per categories.

Label Encoding: In order for machine learning models—which can only accept numerical data—to fit categorical columns, a technique known as label encoding is employed to transform them into numerical ones. In a machine-learning project, it is a crucial pre-processing phase (Chugh, 2018). It is similar to binary classification. Maximum reviewed works (Aksu and Aydin, 2018; Bahaghighat et al., 2023; Innab et al., 2024; Mosa et al., 2023; Shetty et al., 2023) used binary classification. Their dataset contains two types of data. But our dataset contained 4 types. That's why we used label encoding. This work applied label encoding on the 'type' column. After label encoding, 0 is for safe links, 1 is for defacement, 2 is for malware, and 3 is for phishing. **Table 1** represents the picture after applying label encoding.

Table 1. Label encoding of type.

Type	Type Code
safe links	0
defacement	1
malware	2
phishing	3

Features Engineering: The process of adding new features or altering current ones in order to enhance a machine-learning model's performance is known as feature engineering. It entails picking pertinent information from unprocessed data and putting it in a manner that a model can understand. The objective is to increase the model's accuracy by offering more pertinent and useful data (Turner et al., 1999). Only one work (FBI, 2023) used feature engineering techniques. Because their dataset is similar to ours. We used advanced feature engineering with many features. Features engineering was applied to the dataset to add new features. It improved the quality of information. It helped to make a system that can identify the malware website URLs. **Figure 4** contains the diagram of feature engineering.

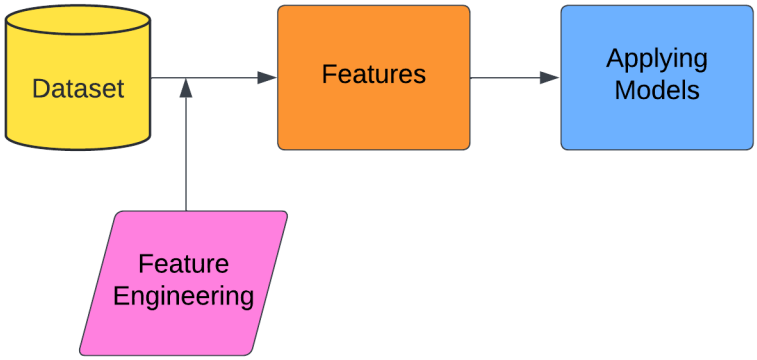

Figure 4. Feature engineering.

A check Ip column was added to the dataset, which helped identify whether it contained Ip address. Abnormal URL were added to identify the valid URLs. Count of dots(.) column stored the calculated dots that each URL contained. It will help to find more dangerous website URLs. More than two dots increase the risk of cyber-attacks. Count of (www.) column contains the count of (www.). A website containing (www.) will be more secure. If the URL has a @ sign, it should be dangerous. That's why the @ sign count was created. Count of HTTP, HTTPS, URL Depth, embed domain, letter, digit, and different signs (%, −, =) were added to detect secure website URLs. URL length, top-level domain (TLD) length, and hostname length were added to the features. All added features made the dataset strong enough to build a system to detect malware of dangerous website URLs.

3.3. Data splitting

The dataset has been divided into 80: 20 ratios. In other words, 20% of the dataset is made up of the testing set, and 80% is made up of the training set. After splitting, the train set contains 520,952 URLs, and the test set has 130,239 URLs. Then, the test

set was divided into a test and a validation set. **Figure 5** shows the pie chart of the train, test, and validation split.

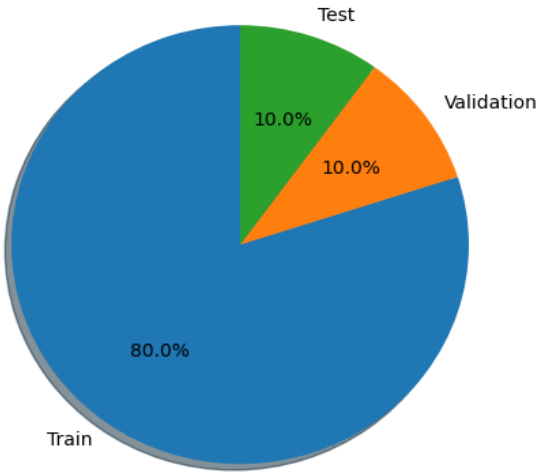

Figure 5. Train, test, and validation ratio.

3.4. Machine learning algorithms

LightGBM: A gradient boosting technique called LightGBM is an ensemble learning framework that builds a strong learner by gradually adding insufficient knowledge in a gradient descent fashion. Using methods like GOSS, it maximizes training time and memory utilization (Ke et al., 2017). LightGBM is used in one work (Shetty et al., 2023). The basic configurations known as LightGBM Core Parameters control how LightGBM models behave and function throughout training. These parameters govern the model's structure, optimization procedure, and goal function, among other things. Fundamental parameters are necessary to adjust the behavior and performance of the model to fit particular machine-learning tasks. The learning rate, number of leaves, maximum depth, regularization terms, and optimization techniques are a few examples of key parameters. It is essential to comprehend and adjust these settings in order to have the best model performance possible while using LightGBM (Wang and Wang, 2020). Whereas another algorithm develops trees horizontally, Light GBM grows vertically, which means that Light GBM expands trees' leaf-wise, whereas the other method develops level-wise. It will select the leaf with maximum delta loss. A leaf-wise approach can reduce loss far more than a level-wise approach when expanding the same leaf (Mandot, 2017). **Figures 6** and **7** show two types of tree growth of Light GBM.

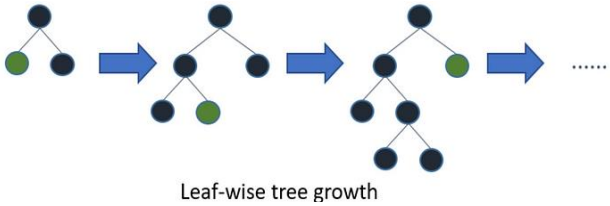

Figure 6. Leaf wise tree growth (Mandot, 2017).

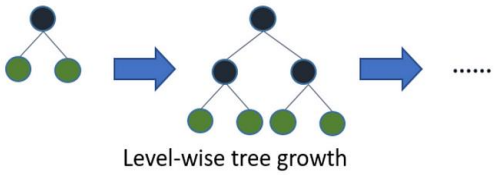

Figure 7. Level wise tree growth (Mandot, 2017).

XGBoost: A networked gradient boosting library intended for efficiency and scalability in machine learning model training is called XGBoost. It is an ensemble method of learning that generates a more powerful prediction by aggregating the predictions of several weak models. Extreme Gradient Boosting, or XGBoost, is a machine learning algorithm that has gained popularity and widespread usage because it can handle large datasets while achieving state-of-the-art efficiency for several machine learning algorithms, including regression and classification (GeeksforGeeks, 2024). This model is used by Bahaghighat et al. (2023), Innab et al. (2024), and Shetty et al. (2023). Even though XGBoost performs well in comparison to other gradient-boosting implementations, it can take a long time to operate. It can take days or even hours to finish common activities. Extensive parameter adjustment is also necessary when employing gradient boosting to build extremely accurate models. The method must be performed numerous times to investigate the impact of variables like the rate of learning and L1/L2 normalization parameters on cross-validation accuracy (Mitchell and Frank, 2017).

Gradient Boosting: Gradient Boosting is a strong boosting technique that turns multiple weak learners into powerful learners as shown in **Figure 8**. It uses the gradient descent technique to teach each new model how to lower the loss functions of its predecessor, such as the mean square error or cross-entropy. In each succeeding iteration, the method determines how the variance function's slope changes about the current ensemble's expectations and then trains a new, subpar model to lessen this gradient. The new model's predictions are then added to the groups, and the cycle continues until an interruption threshold is reached (Bentéjac et al., 2021). No reviewed work used this model. In learning ensemble modeling, one popular technique for building solid classifiers from a range of weak classifiers is "boosting." Using the supplied training data sets, it first constructs a prominent model and then identifies the flaws in the foundational model. After the fault has been identified, a second model is built; a third model is added throughout this process. This process of adding more models is continued until we have a complete training set that the model can accurately predict (Konstantinov and Utkin, 2021).

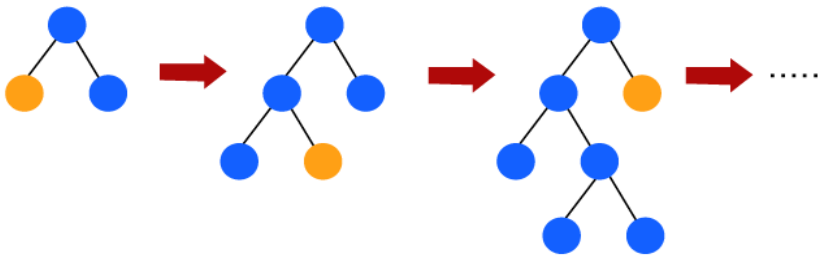

Figure 8. Gradient boosting classifier (Breiman, 2001).

Random Forest: Applications using decision trees, such as regression and classification, make use of the supervised machine learning method known as random forest as shown in **Figure 9**. Using random forests, using vast, complicated datasets, organizing multidimensional feature spaces, and determining the relative relevance of different attributes are all made possible. Numerous industries, including banking, medicine, and image analysis, use this technique because it may minimize overfitting while maintaining high projected accuracy (Breiman, 2001). Random forest used by Bahaghighat et al. (2023), Innab et al. (2024), Lee et al. (2020), and Shetty et al. (2023). A random subset of the training data is chosen using the random forest classification algorithm, producing several decision trees. As the first collection, a random selection of decision trees is taken from the training set. The final prediction is obtained by adding together the votes from each decision tree. Fortunately, combining a sweeping classifier with a decision tree is no longer necessary when utilizing the predictor class of random forest. Random forest can also address regression issues by utilizing the technique's regressor. As trees grow in size, random forest introduces more unpredictability into the model. When dividing a node, it looks for the most beneficial feature from a randomly selected set of features rather than concentrating on the most crucial one. The model gets significantly more diversified and usually better as a result. Because of this, when dividing a node in a random forest classifier, only a random subset of the features is considered (Gunay, 2023).

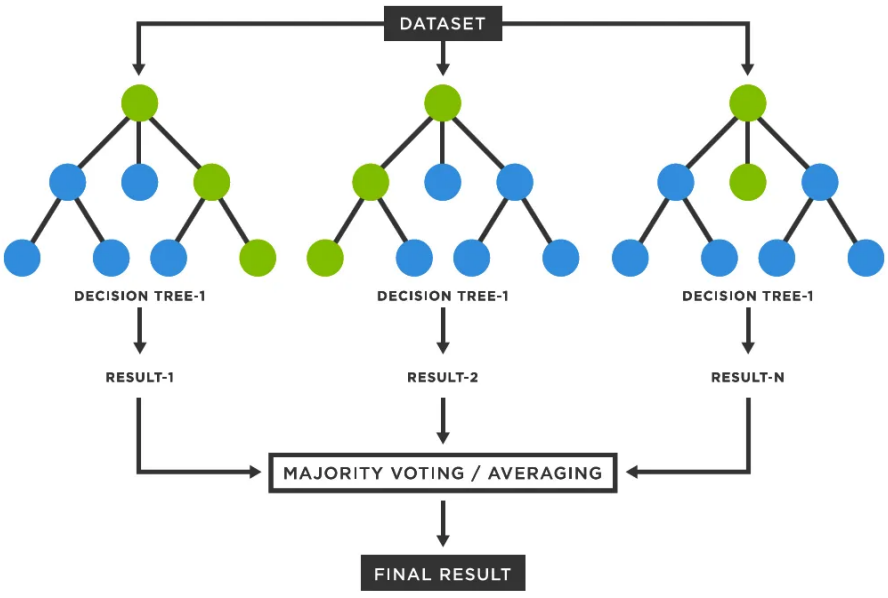

Figure 9. Random forest classifier (Breiman, 2001).

4. Results and discussions

This project used machine learning techniques to detect malware website URLs for cybersecurity. We extracted 651,191 URLs from the Kaggle dataset. To increase the quality of data, feature engineering was used after EDA. Important features were added by using feature engineering to enhance the detection level. Features that weren't needed were removed. Next, random forest, gradient boosting, XGBoost, and Light GBM were applied to the dataset to obtain better outcomes. **Table 2** represents

the result of four models.

Table 2. Four classifier's results.

Model	Accuracy	F1-macro	F1-micro	F1 weighted
Random Forest	0.96	0.95	0.96	0.96
Gradient Boosting	0.94	0.89	0.94	0.93
XGBoost	0.96	0.94	0.96	0.96
Light GBM	0.96	0.94	0.96	0.96

From **Table 2,** the results of random forest, gradient boosting, XGBoost, and Light GBM are represented. The lowest accuracy was secured by a gradient boosting algorithm, which was 94%. The other three models achieved the highest accuracy, 96%. However, according to other scores like F1-macro, F1-micro, and F1 weighted, the random forest gained the highest outcomes. **Figure 10** shows a clear picture of comparing all models.

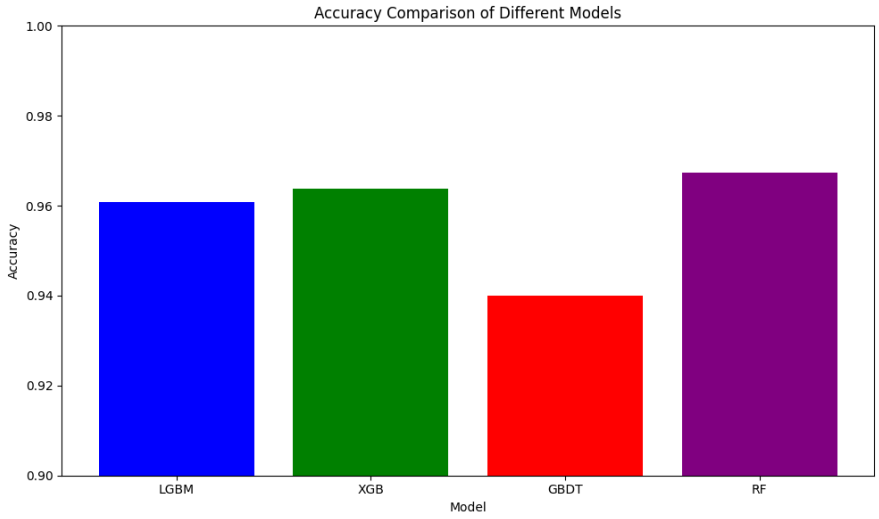

Figure 10. Comparison of four models.

Some important features helped to obtain this outstanding result. The five most important features, which helped most to predict the target result, were hostname_length, count_dir, count of (www.), tld_length, and fd_length. The feature named hostname_length performed well compared to others. **Figure 11** shows the five important features that helped obtain a good outcome.

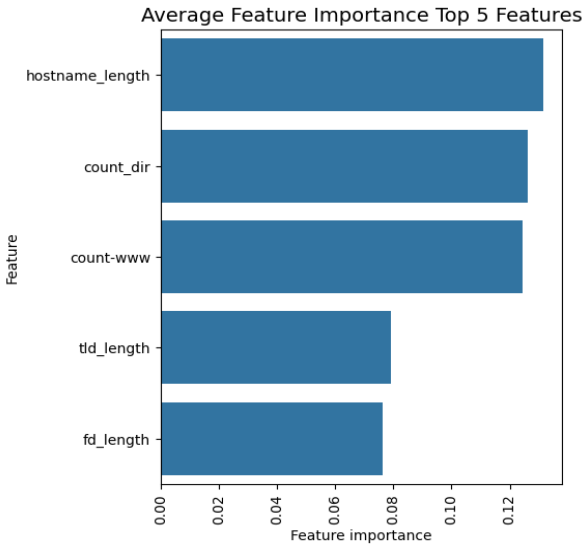

Figure 11. Five important features.

4.1. Confusion matrix

The performance of an artificial intelligence classification algorithm can be precisely represented using a confusion matrix. Totals for false positives, true positives, false negatives, and real positives are included. To summarize the effectiveness of a classification model, a confusion matrix is a tabular representation that contrasts the expected and actual labels. It displays the proportion of TP, FN, FP, and TN predictions that the model made. By improving prediction accuracy and revealing inaccurate classifications, this matrix makes it possible to evaluate the model's performance. A confusion matrix is an $N \times N$ matrix (where N is the total number of target classes) used to analyze the performance of a classification model. The target values in the matrix that actually exist are compared to the expected values of the artificial intelligence model. This gives us a thorough picture of all the different kinds of errors and performance indicators related to our classification model. The confusion matrix is shown in **Figure 12** using the random forest model. Four targeted variables are shown with predicted outcomes.

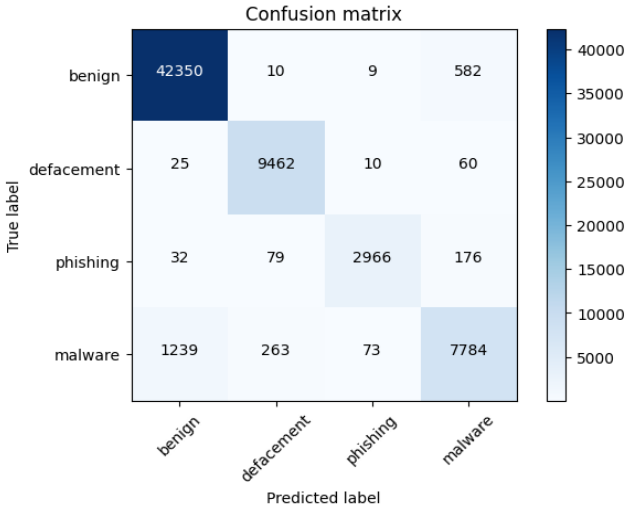

Figure 12. Confusion matrix.

This is a multi-class classification. The four classes are defacement, phishing, benign, and malware. In benign, the model performed well, and 42,350 were true positives. The number of misclassifications is low in this case. 9462 are true positives in defacement, 2966 true positives are in phishing, and lastly, 7784 are true positives in Malware. Misclassifications are low in all classes. The model performed well in every class.

4.2. Accuracy

Optimization of this metric is challenging due to its discrete nature. The frequency of correct result predictions made by a machine learning model is determined by its accuracy. Divide the total estimates by the number of precise forecasts to arrive at the result. Assuming that every class is equally relevant, the accuracy statistic can be used to characterize the model's performance in each one. Larger values imply better model performance. A report on the random forest model's categorization may be found in **Figure 13**.

	precision	recall	f1-score	support
0	0.97	0.99	0.98	42951
1	0.98	1.00	0.99	9557
2	0.99	0.95	0.97	3253
3	0.91	0.86	0.89	9359
accuracy			0.97	65120
macro avg	0.96	0.95	0.96	65120
weighted avg	0.97	0.97	0.97	65120

accuracy: 0.967

Figure 13. Classification report.

For class 0, the model predicted 97% correct predictions. 99% correctly detect actual class 0. On the other hand, the precision rate is 98% and the recall is 100% for class 1. The F1 score is 0.98 for class 0, and 0.99 is in class 0. The model performed with strong accuracy in both classes and the overall accuracy is 97%.

Table 3 presents the comparison of results obtained for different categories of accuracy and performance across the datasets. Kaggle and different datasets are the prominent dataset for identifying cyber-attacks and most of the research utilized it. Malicious URL dataset is a unique dataset that we used in this work. Feature engineering implemented to the dataset to enrich the quality of data. In the feature engineering, URL length, TLD length, hostname length, count of dots, count of different signs, count of dot com, count of www. and other important features added to the dataset, which is the novelty of this work. It increases the quality of each data and detect ability. Kaggle (Malicious URLs dataset), which is a new and unique dataset, was employed to acquire a good result and we successfully obtained 96% accuracy using a random forest classifier. In others research, there is no one who used this dataset. It's made this work unique. **Table 3** offers a decent picture of comparison.

Table 3. Comparison of this work with existing systems.

Author	Dataset	Network	Accuracy
(Shetty et al., 2023)	Kaggle	Random Forest	–
(Innab et al., 2024)	Two phishing dataset	XBG	0.97
		All Models	1.00
(Mosa et al., 2023)	Kaggle	AdaBoost	0.95
(Bahaghighat et al., 2023)	Phishing Websites Dataset	XGBoost	0.99
(Shahriar et al., 2020)	Kaggle	logistic regression	0.99
(Chen et al., 2020)	Kaggle (Microsoft Malware Classification Challenge)	SVM with ALBL	–
(Hassan et al., 2019)	ES	ELK stack with unsupervised learning	–
(Aksu and Aydin, 2018)	CICIDS2017	Deep learning	0.97
(Z. S. Lee et al., 2020)	A legal source	KNN	1.00
(Zhao et al., 2020)	UCI	AdaBoost	0.98
(Lee et al., 2020)	A legal source	SVM with SGD	94%
This Work	Kaggle (Malicious URLs dataset)	Random Forest	0.96

5. Conclusions

Finally, malware URLs that are involved in cyber-attacks were detected through machine learning algorithms. From the Kaggle dataset, 651,191 URLs were extracted. To increase the quality of data, feature engineering was used after EDA. Important features were added to the collected dataset using feature engineering to enhance the detection level, making us unique from others. Features that weren't needed were removed. Next, four models are used to get a better outcome using the collected dataset. To train our models, we used advanced data preprocessing, data cleaning, and well-known classifiers in our work. The dataset was trained using random forest, gradient boosting, XGBoost, and Light GBM, which helped to get good results using the dataset. We used multiple preparation techniques to ensure that our dataset was noise-free. Feature engineering and label encoding helped a lot to increase the quality of data and prevented overfitting. From four models, the random forest gained a better F1 score than others. Whether one is superior to the other, we used Kaggle (Malicious URLs dataset) with feature engineering to gain a good result, and we successfully obtained 96% accuracy using the random forest classifier. We can use different types of hyper tuning. We can use different types of hyper tuning to improve the model performance to improve the model performance. If we change the parameters of hyperparameters of models, it will give a better outcome. In the future, we will strengthen the accuracy of the models employed in this study. We will additionally incorporate or train more new NLP models to develop a project with improved accuracy and enhanced cybersecurity.

Author contributions: Conceptualization, SFF, BB, MIS and RH; methodology, NM, MZA and MA; software, SFF, MIS, MZA and RH; validation, NM, SFF, BB and MA; formal analysis, SFF, BB, MZA and MA; investigation, SFF, NM and RH;

resources, BB, MIS, MZA and RH; data curation, SFF, BB, MZA and MA; writing—original draft preparation, SFF, BB, MIS, NM, MZA and RH; writing—review and editing, SFF, BB, MIS, MZA, RH and MA; visualization, SFF, NM, MZA and MA; supervision, SFF, RH and MA; project administration, RH and MA; funding acquisition, MA. All authors have read and agreed to the published version of the manuscript.

Acknowledgments: The authors would like to thank the University of Debrecen Program for Scientific Publication for the research support.

Conflict of interest: The authors declare no conflict of interest.

References

Aksu, D., & Aydin, M. A. (2018). Detecting Port Scan Attempts with Comparative Analysis of Deep Learning and Support Vector Machine Algorithms. 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT).

Akter, J., Nilima, S. I., Hasan, R., Tiwari, A., Ullah, M. W., & Kamruzzaman, M. (2024). Artificial intelligence on the agro-industry in the United States of America. AIMS Agriculture and Food, 9(4), 959–979. https://doi.org/10.3934/agrfood.2024052

Bahaghighat, M., Ghasemi, M., & Ozen, F. (2023). A high-accuracy phishing website detection method based on machine learning. Journal of Information Security and Applications, 77, 103553. https://doi.org/https://doi.org/10.1016/j.jisa.2023.103553

Bentéjac, C., Csörgő, A., & Martínez-Muñoz, G. (2021). A comparative analysis of gradient boosting algorithms. Artificial Intelligence Review, 54(3), 1937–1967. https://doi.org/10.1007/s10462-020-09896-5

Bhuyan, M. K., Kamruzzaman, M., Nilima, S. I., Khatoon, R., & Mohammad, N. (2024). Convolutional Neural Networks Based Detection System for Cyber-attacks in Industrial Control Systems. Journal of Computer Science and Technology Studies, 6(3), 86–96.

Biswas, B., Sharmin, S., Hossain, M. A., Alam, M. Z., & Sarkar, M. I. (2024). Risk Analysis-based Decision Support System for Designing Cybersecurity of Information Technology. Journal of Business and Management Studies, 6(5), 13–22.

Breiman, L. (2001). Random forests. Machine learning, 45, 5–32.

Buss, D. D. (2002). Technology in the Internet age. 2002 IEEE International Solid-State Circuits Conference. Digest of Technical Papers (Cat. No.02CH37315).

Chen, C. W., Su, C. H., Lee, K. W., & Bair, P. H. (2020). Malware Family Classification using Active Learning by Learning. 2020 22nd International Conference on Advanced Communication Technology (ICACT).

Chowdhury, A. A. M., & Arefeen, S. (2011). Software risk management: importance and practices. IJCIT, ISSN, 2078-5828.

Chugh, A. (2018). Label Encoding of datasets in Python. GeeksforGeeks.

Datta, S. D., Islam, M., Rahman Sobuz, M. H., Ahmed, S., & Kar, M. (2024). Artificial intelligence and machine learning applications in the project lifecycle of the construction industry: A comprehensive review. Heliyon, 10(5), e26888. https://doi.org/https://doi.org/10.1016/j.heliyon.2024.e26888

Datta, S. D., Sarkar, M. M., Rakhe, A. S., Aditto, F. S., Sobuz, M. H. R., Shaurdho, N. M. N., Nijum, N. J., & Das, S. (2024). Analysis of the characteristics and environmental benefits of rice husk ash as a supplementary cementitious material through experimental and machine learning approaches. Innovative Infrastructure Solutions, 9(4), 121. https://doi.org/10.1007/s41062-024-01423-7

Federal Bureau of Investigation (FBI). (2023). Cyber Crime | Federal Bureau of Investigation," Federal Bureau of Investigation. Retrieved 5 July 2024 from https://www.fbi.gov/investigate/cyber

Fortinet. (2024). 7 Common Web Security Threats for an Enterprise. Retrieved 15 July 2024 from https://www.fortinet.com/resources/cyberglossary/web-security-threats#:~:text=The%20most%20common%20web%20security

Gallagher, S., & Brandt, A. (2020). Facing down the myriad threats tied to COVID-19. Sophos, Abingdon, United Kingdom. Available online: https://news. sophos. com/en-us/2020/04/14/covidmalware/(accessed on January 2024).

GeeksforGeeks. (2024). XGBoost. . Retrieved 8 August 2024 from https://www.geeksforgeeks.org/xgboost/

Görgün, E. (2022). Characterization of Superalloys by Artificial Neural Network Method. Online International Symposium on Applied Mathematics and Engineering (ISAME22) January 21–23, 2022 Istanbul-Turkey,

Gunay, D. (2023). Random Forest. Medium.

Habibur Rahman Sobuz, M., Khan, M. H., Kawsarul Islam Kabbo, M., Alhamami, A. H., Aditto, F. S., Saziduzzaman Sajib, M., Johnson Alengaram, U., Mansour, W., Hasan, N. M. S., Datta, S. D., & Alam, A. (2024). Assessment of mechanical properties with machine learning modeling and durability, and microstructural characteristics of a biochar-cement mortar composite. Construction and Building Materials, 411, 134281. https://doi.org/https://doi.org/10.1016/j.conbuildmat.2023.134281

Hasan, N. M. S., Sobuz, M. H. R., Shaurdho, N. M. N., Meraz, M. M., Datta, S. D., Aditto, F. S., Kabbo, M. K. I., & Miah, M. J. (2023). Eco-friendly concrete incorporating palm oil fuel ash: Fresh and mechanical properties with machine learning prediction, and sustainability assessment. Heliyon, 9(11). https://doi.org/10.1016/j.heliyon.2023.e22296

Hassan, A., Tahir, S., & Baig, A. I. (2019). Unsupervised Machine Learning for Malicious Network Activities. 2019 International Conference on Applied and Engineering Mathematics (ICAEM),

Hossain, M. A., Tiwari, A., Saha, S., Ghimire, A., Imran, M. A. U., & Khatoon, R. (2024). Applying the Technology Acceptance Model (TAM) in Information Technology System to Evaluate the Adoption of Decision Support System. Journal of Computer and Communications, 12(8), 242–256.

Innab, N., Osman, A. A. F., Ataelfadiel, M. A. M., Abu-Zanona, M., Elzaghmouri, B. M., Zawaideh, F. H., & Alawneh, M. F. (2024). Phishing Attacks Detection Using Ensemble Machine Learning Algorithms. Computers, Materials and Continua, 80(1), 1325–1345. https://doi.org/https://doi.org/10.32604/cmc.2024.051778

Jamil, A., Asif, K., Ashraf, R., Mehmood, S., & Mustafa, G. (2018). A comprehensive study of cyber attacks & counter measures for web systems Proceedings of the 2nd International Conference on Future Networks and Distributed Systems, Amman, Jordan. https://doi.org/10.1145/3231053.3231116

Johora, F. T., Hasan, R., Farabi, S. F., Alam, M. Z., Sarkar, M. I., & Mahmud, M. A. A. (2024). AI Advances: Enhancing Banking Security with Fraud Detection. 2024 First International Conference on Technological Innovations and Advance Computing (TIACOMP),

Johora, F. T., Manik, M. M. T. G., Tasnim, A. F., Nilima, S. I., & Hasan, R. (2021). Advanced-Data Analytics for Understanding Biochemical Pathway Models. American Journal of Computing and Engineering, 4(2), 21–34.

Kamruzzaman, M., Bhuyan, M. K., Hasan, R., Farabi, S. F., Nilima, S. I., & Hossain, M. A. (2024). Exploring the Landscape: A Systematic Review of Artificial Intelligence Techniques in Cybersecurity. 2024 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI).

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems, 30.

Khan, M. M. H., Sobuz, M. H. R., Meraz, M. M., Tam, V. W. Y., Hasan, N. M. S., & Shaurdho, N. M. N. (2023). Effect of various powder content on the properties of sustainable self-compacting concrete. Case Studies in Construction Materials, 19, e02274. https://doi.org/https://doi.org/10.1016/j.cscm.2023.e02274

Khan, N. A., Brohi, S. N., & Zaman, N. (2023). Ten deadly cyber security threats amid COVID-19 pandemic. Authorea Preprints.

Konstantinov, A. V., & Utkin, L. V. (2021). Interpretable machine learning with an ensemble of gradient boosting machines. Knowledge-Based Systems, 222, 106993. https://doi.org/https://doi.org/10.1016/j.knosys.2021.106993

Lee, Roh, H., & Lee, W. (2020). Poster Abstract: Encrypted Malware Traffic Detection Using Incremental Learning. IEEE INFOCOM 2020—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS),

Lee, Z. S., Guo, H., & Zhou, L. (2020). Rail System Anomaly Detection via Machine Learning Approaches. 2020 IEEE REGION 10 CONFERENCE (TENCON).

Linkon, A. A., Noman, I. R., Islam, M. R., Bortty, J. C., Bishnu, K. K., Islam, A., Hasan, R., & Abdullah, M. (2024). Evaluation of Feature Transformation and Machine Learning Models on Early Detection of Diabetes Melitus. IEEE Access, 1-1. https://doi.org/10.1109/ACCESS.2024.3488743

Mandot, P. (2017). What is LightGBM, How to implement it? How to fine tune the parameters. Online. In Medium. https://medium. com/@ pushkarmandot/https-medium-com-pushkarmandotwhat-is-lightgbm-how-to-implement-it-how-to-fine-tune-the-parameters-60347819b7fc.

Manik, M. M. T. G., Nilima, S. I., Mahmud, M. A. A., Sharmin, S., & Hasan, R. (2022). Discovering Disease Biomarkers in Metabolomics via Big Data Analytics. American Journal of Statistics and Actuarial Sciences, 4(1), 35–49. https://doi.org/https://doi.org/10.47672/ajsas.2452

Mehedi, M. T., Sobuz, M. H. R., Hasan, N. M. S., Jabin, J. A., Nijum, N. J., & Miah, M. J. (2024). High-strength fiber reinforced concrete production with incorporating volcanic pumice powder and steel fiber: sustainability, strength and machine learning technique. Asian Journal of Civil Engineering. https://doi.org/10.1007/s42107-024-01169-8

Mitchell, R., & Frank, E. (2017). Accelerating the XGBoost algorithm using GPU computing. PeerJ Computer Science, 3, e127.

Mohammad, N., Khatoon, R., Nilima, S. I., Akter, J., Kamruzzaman, M., & Sozib, H. M. (2024). Ensuring Security and Privacy in the Internet of Things: Challenges and Solutions. Journal of Computer and Communications, 12(8), 257–277.

Mosa, D. T., Shams, M. Y., Abohany, A. A., El-kenawy, E.-S. M., & Thabet, M. (2023). Machine Learning Techniques for Detecting Phishing URL Attacks. Computers, Materials and Continua, 75(1), 1271–1290. https://doi.org/https://doi.org/10.32604/cmc.2023.036422

Mottakin, M., Datta, S. D., Hossain, M. M., Sobuz, M. H. R., Rahman, S. M. A., & Alharthai, M. (2024). Evaluation of textile effluent treatment plant sludge as supplementary cementitious material in concrete using experimental and machine learning approaches. Journal of Building Engineering, 96, 110627. https://doi.org/https://doi.org/10.1016/j.jobe.2024.110627

Nilima, S. I., Bhuyan, M. K., Kamruzzaman, M., Akter, J., Hasan, R., & Johora, F. T. (2024). Optimizing Resource Management for IoT Devices in Constrained Environments. Journal of Computer and Communications, 12(8), 81–98.

Shahriar, H., Qian, K., & Zhang, H. (2020). Learning Environment Containerization of Machine Leaning for Cybersecurity. 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC),

Shetty, Patil, A., & Mohana. (2023). Malicious URL Detection and Classification Analysis using Machine Learning Models. 2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT),

Siddhartha, M. (2024). Malicious URLs dataset. Retrieved 8 June 2024 from https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset/data

Sobuz, M. H. R., Aditto, F. S., Datta, S. D., Kabbo, M. K. I., Jabin, J. A., Hasan, N. M. S., Khan, M. M. H., Rahman, S. M. A., Raazi, M., & Zaman, A. A. U. (2024). High-Strength Self-Compacting Concrete Production Incorporating Supplementary Cementitious Materials: Experimental Evaluations and Machine Learning Modelling. International Journal of Concrete Structures and Materials, 18(1), 67. https://doi.org/10.1186/s40069-024-00707-7

Sobuz, M. H. R., Al, I., Datta, S. D., Jabin, J. A., Aditto, F. S., Sadiqul Hasan, N. M., Hasan, M., & Zaman, A. A. U. (2024). Assessing the influence of sugarcane bagasse ash for the production of eco-friendly concrete: Experimental and machine learning approaches. Case Studies in Construction Materials, 20, e02839. https://doi.org/https://doi.org/10.1016/j.cscm.2023.e02839

Sobuz, M. H. R., Jabin, J. A., Ashraf, J., Anzum, M. T., Shovo, A. R., Rifat, M. T. R., & Adnan, T. (2024). Enhancing sustainable concrete production by utilizing fly ash and recycled concrete aggregate with experimental investigation and machine learning modeling. Journal of Building Pathology and Rehabilitation, 9(2), 134.

Sobuz, M. H. R., Joy, L. P., Akid, A. S. M., Aditto, F. S., Jabin, J. A., Hasan, N. M. S., Meraz, M. M., Kabbo, M. K. I., & Datta, S. D. (2024). Optimization of recycled rubber self-compacting concrete: Experimental findings and machine learning-based evaluation. Heliyon, 10(6), e27793. https://doi.org/https://doi.org/10.1016/j.heliyon.2024.e27793

Swathi, Y., Hegde, P., Sravani, P., & Hegde, P. (2023). Detection of Phishing Websites Using Machine Learning. 2023 International Conference on the Confluence of Advancements in Robotics, Vision and Interdisciplinary Technology Management (IC-RVITM).

Tang, L., & Mahmoud, Q. H. (2021). A Survey of Machine Learning-Based Solutions for Phishing Website Detection. Machine Learning and Knowledge Extraction, 3(3), 672–694.

Turner, C. R., Fuggetta, A., Lavazza, L., & Wolf, A. L. (1999). A conceptual basis for feature engineering. Journal of Systems and Software, 49(1), 3–15.

Wang, Y., & Wang, T. (2020). Application of Improved LightGBM Model in Blood Glucose Prediction. Applied Sciences, 10(9).

Zhao, Q., Sun, J., Ren, H., & Sun, G. (2020). Machine-Learning Based TCP Security Action Prediction. 2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE).