Malicious URL Classification with Instance Selection

Xavier Fulkerson  
 Computer and Software Systems Department  
 University of Washington Bothell

Dong Si  
Computer and Software Systems Department  
 University of Washington Bothell

ABSTRACT

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

© 2018 Copyright held by the owner/author(s).

The modern internet uses hypertext with URLs to enable seamless navigation for end users. However, malicious third-parties abuse this hypertext design of the internet to access, steal, or otherwise illegitimately obtain user data. Many malicious URLs are becoming indistinguishable to end users when compared to benign URLs, necessitating new, innovative security measures to protect end users from these bad actors. Leveraging feature extraction, instance selection and machine learning, our approach achieves accuracy of machine learning on par with existing research using instance selection with fractional training times. Our contributions extend to enhancing the efficiency of malicious URL identification models, thereby improving real-time cybersecurity interception capabilities.

CCS CONCEPTS

•Security and privacy~Systems security~Browser security•Security and privacy~Network security~Web protocol security

KEYWORDS

cybersecurity, malicious URL, machine learning, lexical analysis, feature extraction, instance selection

1 Introduction

As of 2023, the worldwide internet user base has surpassed 5.18 billion, consisting of nearly two-thirds of the world’s population [1]. In this modern digital landscape, protecting end users against malicious third parties is an ever more relevant task in the field of cybersecurity. The massive scale of the internet enables bad actors to compromise user data, utilizing malicious URLs to enact phishing, malware, and defacement attacks. As end users become more vigilant, cybercriminals continue to employ increasingly sophisticated methods to obfuscate the URLs used to carry out these attacks, underscoring the pressing need for enhanced security measures against malicious URLs.  
 While existing solutions, such as dynamic blacklists, aim to catalog known malicious URLs [2], the practicality of implementing these solutions onto end-user devices directly proves to be unrealistic due to the bandwidth constraints of individual users and the time spent discovering and identifying new malicious URLs. Addressing these challenges, this project aims to improve the methods utilized in threat identification through the use of machine learning (ML) models. Specifically, we explore the efficacy of gradient boosting ML models with random instance selection. This research is motivated by the goal of enhancing the model training time to explore feasible options of implementing robust solutions for identifying malicious URLs directly in end-user devices.

2 Related work

Previous research, such as that conducted by Aljabri et. al.[3], has extensively analyzed various aspects of URLs for malicious URL detection. Aljabri defined the three main components: Lexical-based, Content-based, and Network-based. Lexical components can be analyzed by tokenizing the URL string, content components can be found by analyzing the content of the webpage, and network components can be found by analyzing network activity like latency and packet size. For the purpose of user-focused design, this algorithm will only consider lexical-based analysis of URLs to prevent security risks involved with contacting malicious URLs to perform content and network-based analysis.

Aljabri et. al. demonstrated the effectiveness of Naïve Bayes (NB) for identifying malicious URLs, achieving a 95% success rate by training ML models with all three components of malicious URL detection [3]. However, this study does not explore the optimizations offered by feature selection, causing considerably long training times, and implementing the machine requires the end-user to put their device at risk to content and network-based analysis on a URL to determine the risk involved with clicking it.

Upendra Shetty et. al. explored the use of XGB, LGB, and Random Forests (RF) with a dataset of over 650,000 URLs categorized into malware, defacement, benign, and phishing, finding RF outperformed the other models at a rate of over 91% [4]. Another recent study conducted by Abad et. al. used the same dataset to study the potential of using instance selection to determine the efficacy of RFs, Decision Trees (DTs), Support Vector Machines (SVMs), and K-Nearest Neighbors (KNNs) with Bayesian optimization to classify URLs, finding that random instance selection with RFs yielded the highest F1 score of 92% [5]. However, both of these studies utilized a Kaggle dataset that contained over 90,000 mislabeled records, where phishing and benign labels were switched when the dataset was aggregated. Additionally, the dataset contained fewer malware links compared to other link types, leading to an imbalance in the data.

In our study, we aim to identify the potential benefits of employing XGB and LGB with instance selection using a fixed version of the Kaggle dataset augmented with additional malware links.

3 Method

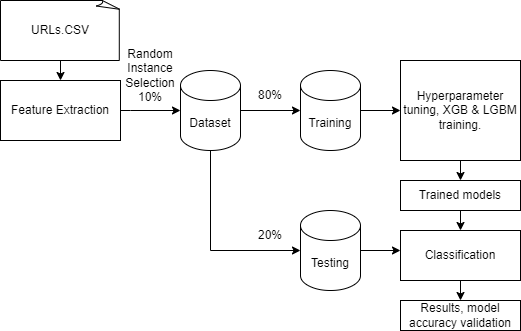


Figure 1: Diagram representation of dataset processing and model training

The methodology employed within this study aims to address the challenges when faced with malicious URL detection on end-user systems through a combination of feature extraction, instance selection, and machine learning. This approach is designed to improve the accuracy and efficiency of the detection process. **Figure 1** provides an overview of the approach used to identify malicious URLs in this project.

Our dataset utilizes a collection of over 660,000 URLs from the Malicious URLs Kaggle dataset[6]. Additionally, the improperly labeled section retrieved from PhishStorm[7] was fixed, and additional malware URLs from URLhaus was retrieved to augment the dataset. To ensure the quality of the dataset, we performed preprocessing to clean the dataset of improperly formatted entries and repeated entries by manually removing improperly formatted entries and removing repeat entries with a Python script.

**Table 1: URL anatomy**

Features selected for feature analysis

|  |  |
| --- | --- |
| Feature name | Data type |
| has\_IP | boolean |
| has\_HTTP | boolean |
| has\_HTTPS | boolean |
| has\_HTML | boolean |
| has\_PHP | boolean |
| has\_atsign | boolean |
| length | integer |
| domain\_length | integer |
| path\_slash\_count | integer |
| domain\_dot\_count | integer |
| hyphen\_count | integer |
| non\_alphanumeric\_count | integer |

|  |
| --- |
| **Algorithm 1:** Tokenize the URL to generate list of features |
| Input: URL |
| **Output:** Feature array  **Procedure** tokenize(URL)  **Initialize** count = 0  **for all** boolean feature to extract; count++  **if** feature is present  feature[i] = 1  **else** |
| feature[i] = 0  **for all** integer feature to extract; count++  feature[count] = # instances of feature |

To prepare the data for the ML models, the dataset underwent feature extraction. **Table 1** details the features that were selected to identify a URL in the extracted data. Utilizing the research performed by Aljabri et. al.[3] and Upendra Shetty et. al.[4], these features were selected to best represent a URL without overfitting. **Algorithm 1** provides a pseudocode overview of how the features were extracted from the URL to be compiled to another dataset that would be used to train the model.

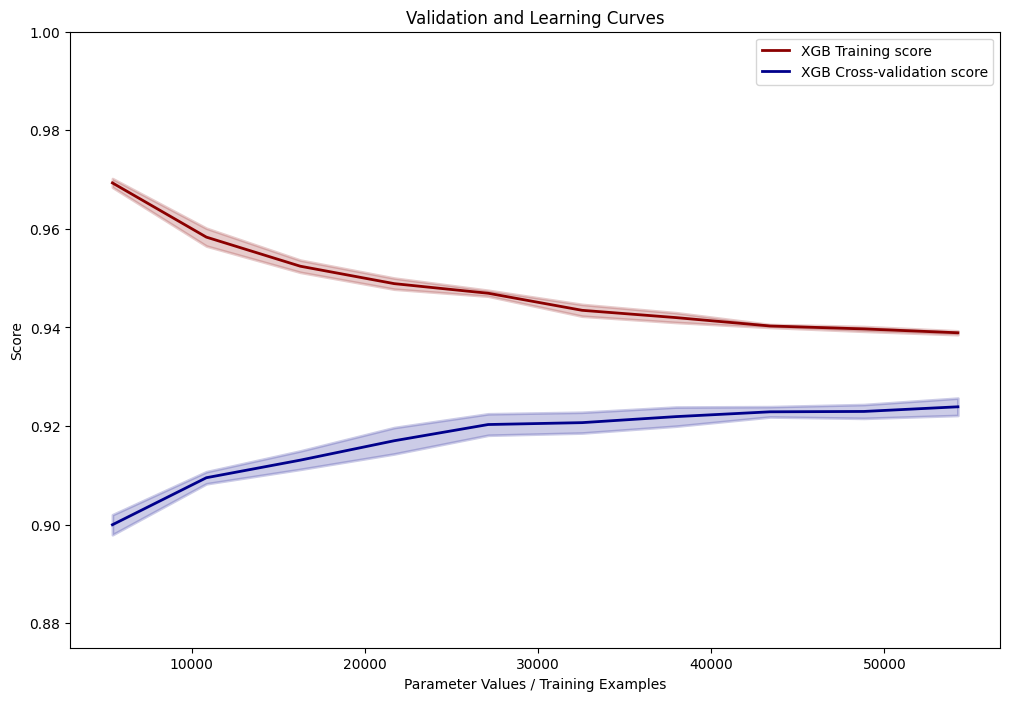
Using the results from Upendra Shetty et. al.[4] that found random instance selection to provide the greatest performance and precision over data reduction based on locality-sensitive hashing and border point extraction based on locality-sensitive hashing, we used random instance selection for this project. The instance selection was implemented such that 10% of the dataset was randomly selected to train each model to reduce computation times to evaluate the prospect of gradient boosting models being used in real-time malicious URL identification on end-user devices. Random selection from the dataset with a better balance of entry types from the augmented malicious URLs is the justification for this method.

We employed XGBoost (XGB) and LightGBM (LGBM) as our machine learning models based on their observed usefulness for malicious URL identification in previous studies [4, 9, 10]. The models underwent hyperparameter optimization by performing a grid search on the dataset obtained from instance selection to find optimal parameters to run the XGB and LGBM models on. XGB was trained using a learning rate of 0.2, a max depth of 7, and 200 estimators. LGBM was trained using a learning rate of 0.1, a max depth of 15, 100 estimators, and 128 leaves.

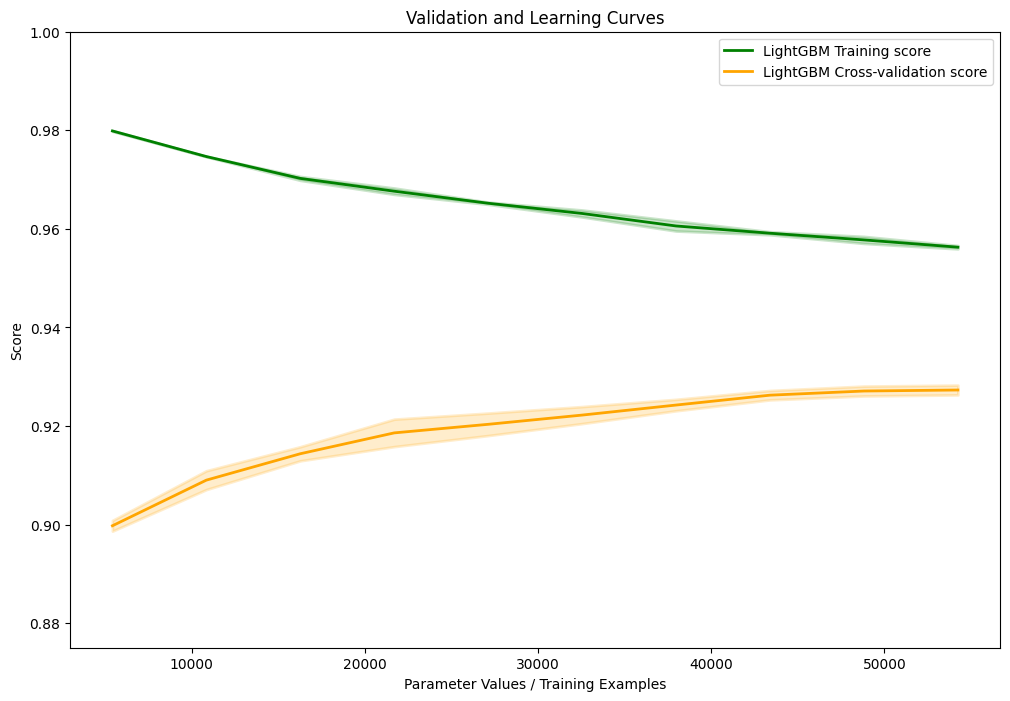
The dataset was divided into training and testing sets using an 80/20 split. The model was trained and evaluated using validation and learning curves, confusion matrices, accuracy, precision, recall, and F1 score to ensure that the model is producing accurate results.

The performance of the model is evaluated using a timer to identify the training time and prediction time of the models. The results of this study may vary based on the hardware used to train the model, which is significant to note when considering the potential applications of these models being used on end-user devices, which may see increases or decreases in performance.

4 Results



**Figure 2: Validation and Learning curves for XGB**



**Figure 3: Validation and Learning curves for LGBM**

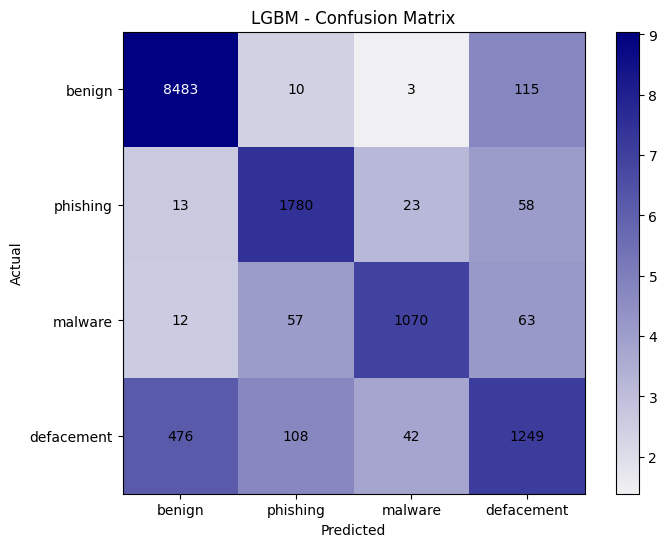
After performing feature extraction and dividing the data into training and testing sets, we trained the models using the methods described in Section 3. The models were originally configured without any modifications, and were altered after undergoing hyperparameter optimization to further increase the accuracy of the model. **Figure 2** and **Figure 3** were used to validate that the model did not undergo overfitting, and to observe the model converging on the best fitting accuracy it could achieve.

**Table 2: Model performance on test set after training**

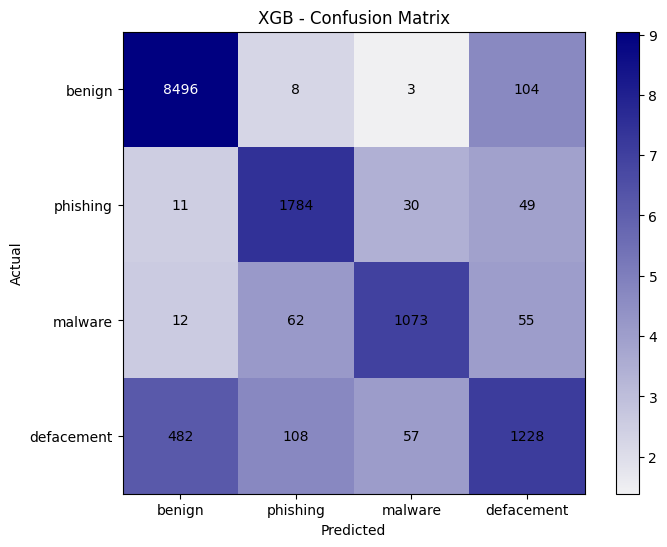
|  |  |  |
| --- | --- | --- |
| Metric | XGB | LGBM |
| Accuracy | 92.8% | 92.8% |
| Precision | 92.5% | 92.5% |
| Recall | 92.8% | 92.8% |
| F1 Score | 92.4% | 92.5% |
| Training Time | 1.132 s | 0.972 s |
| Prediction Time | 0.027 s | 0.067 s |

Using the results displayed in **Table 2**, it can be observed that both XGB and LGBM performed very similarly on the instance selected dataset. The metrics observed from these gradient boosting algorithms have shown to outperform all instance selection algorithms and every model tested by Abad et. al., excluding precision using random instance selection with RF, which had a Precision score of 93.19% [5]. This is a significant result as this shows XGB and LGBM’s efficacy as models trained on a dataset that underwent instance selection for malicious URL identification.

Additionally, the training time of the models outperform every model tested in the study performed by Abad et. al. in training time, with the lowest training time from the study being random instance selection with DT at 16 seconds [5]. This proves that XGB and LGBM are exceptional models for training a dataset quickly, which is relevant for potential real-time applications of this model.

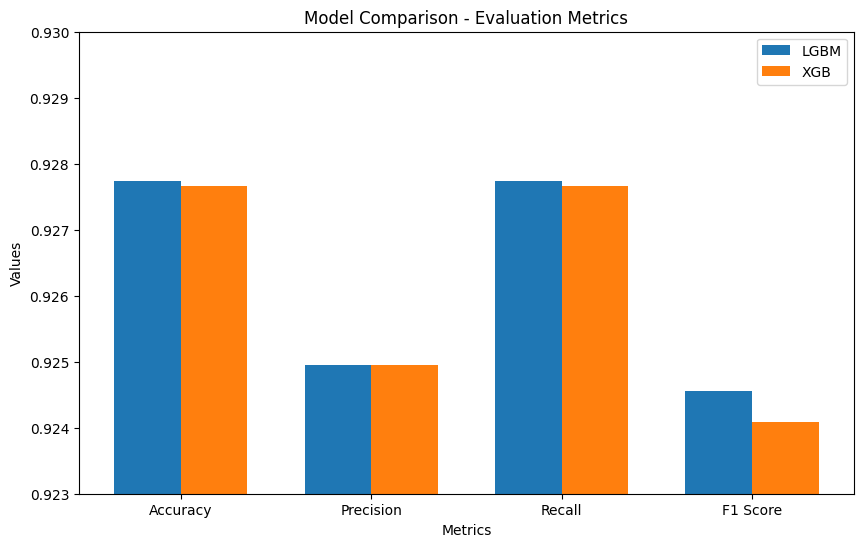


**Figure 4: LGBM Confusion Matrix**

****

**Figure 5: XGB Confusion Matrix**

The confusion matrices for LGBM and XGB show a similar pattern. Benign URLs are relatively well defined for phishing and malware but are not well defined when compared to defacement URLs, having high levels of inaccuracy for both false positives and true negatives. Additionally, phishing URLs show a false positive rate for defacement more readily than other observed malicious URL confusion. The biggest issue with this confusion matrix is the false positive and negative rate of defacement URLs. Typically, these URLs are disguised to appear very similarly to benign URLs, more so than the other URL types, and in some cases are unrecognizably different without network or content analysis [3]. False positives and true negatives are less relevant for malicious URLs as they are still being properly identified to not be opened.



**Figure 6: LGBM and XGB Evaluation Metrics**

**Figure 6** shows the evaluation metrics for LGBM and XGB evaluation metrics side-by-side. Although not apparent from the table, this clearly shows that LGBM exhibited better results than XGB, with the most notable difference being the F1 score. This is significant as LGBM can be considered the better model for identifying malicious URLs with instance selection.

5 Conclusion

In conclusion, this study has explored the application of machine learning with instance selection to identify malicious URLs. After testing XGB and LGBM with instance selection, it is clear that both models show potential for use in real-time applications for malicious URL detection in both their performance and their accuracy, with XGB achieving 92.8% accuracy in 1.132 s training time, and LGBM achieving 92.8% accuracy in 0.972 s training time. These results can be attributed to the robustness of the gradient boosting algorithms used by XGB and LGBM, which can generate accurate predictions with unbalanced datasets in a short period of time relative to other machine learning algorithms.

The testing of these models can be adjusted and fine-tuned to better handle improper identification of defacement URLs. Additional feature engineering should be conducted to address this issue with the model if it is to be used in an end-users system to prevent improper blocking of benign URLs. Additional measures may need to be taken to ensure that no benign websites are identified as false positives for the end-user’s quality of life, as well. We suggest that this model should only be used to warn a user of a potentially dangerous site as opposed to blocking the website outright until the algorithm can be validated to identify no false positives. Additionally, model accuracy can be increased by simply identifying benign and malicious URLs only. Dividing URLs into types of malicious URL may be overcomplicating the process of identifying malicious URLs, and this reduction in complexity may increase the model’s viability as a product for end-users.

LGBM has proven to be effective for identifying malicious URLs, and any additional testing on improving model accuracy is recommended to use LGBM as the primary focus, due to the faster training time and better performance in all metrics over XGB.

REFERENCES

[1] Petrosyan, Ani. (Oct 24, 2023). Topic: Internet usage worldwide. Statista. Retrieved November 8, 2023, from <https://www.statista.com/topics/1145/internet-usage-worldwide/>

[2] *URLhaus | Malware URL exchange.* (n.d.). Retrieved November 8, 2023, from <https://urlhaus.abuse.ch/>

[3] Aljabri, M., Alhaidari, F., Mohammad, R. M. A., Samiha Mirza, U., Alhamed, D. H., Altamimi, H. S., & Chrouf, S. M. B. (2022). 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*, e3241216. <https://doi.org/10.1155/2022/3241216>

[4] R., Upendra Shetty D., 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)*, 470–476. <https://doi.org/10.1109/IDCIoT56793.2023.10053422>

[5] Abad, S., Gholamy, H., & Aslani, M. (2023). Classification of Malicious URLs Using Machine Learning. *Sensors (Basel, Switzerland)*, 23(18). <https://doi.org/10.3390/s23187760>

[6] Siddhartha, Manu. *Malicious URLs dataset.* (n.d.). Retrieved November 8, 2023, from <https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset>Conference Name:ACM Woodstock conference

[7] PhishStorm Phishing / legitimate URL dataset. (n.d.). *Aalto University’s Research Portal*. Retrieved November 8, 2023, from <https://research.aalto.fi/en/datasets/phishstorm-phishing-legitimate-url-dataset>

[8] *URLhaus | Malware URL exchange.* (n.d.). Retrieved November 8, 2023, from <https://urlhaus.abuse.ch/>

[9] He, S., Li, B., Peng, H., Xin, J., & Zhang, E. (2021). *An Effective Cost-Sensitive XGBoost Method for Malicious URLs Detection in Imbalanced Dataset*. In IEEE Access (Vol. 9, pp. 93089–93096). Institute of Electrical and Electronics Engineers (IEEE). <https://doi.org/10.1109/access.2021.3093094>

[10] A, T., & John, A. (2022). *Phishing Website Detection Using LGBM Classifier With URL-Based Lexical Features*. In 2022 IEEE Silchar Subsection Conference (SILCON). 2022 IEEE Silchar Subsection Conference (SILCON). IEEE. <https://doi.org/10.1109/silcon55242.2022.10028793>