Predicting the Legitimacy of URLs

Nicholas Suchy
DATA 450 - Data Mining
Professor Bejarano

I. Introduction

The world's expanding knowledge on technology has brought numerous benefits to society. For example, today, people can manage their bank accounts, purchase items, and even manage their personal health information from their homes via the internet. However, with these conveniences comes the growing concern of cybersecurity threats. Cybercriminals exploit vulnerabilities in systems to steal sensitive information, achieve financial gain, or even just for the acknowledgement [1]. As technology becomes more integrated into everyday life, the need for robust security measures and heightened user awareness has never been more critical.

Of the numerous cybercrimes executed by criminals, phishing is one of the most prominent. Nearly 41% of all cybersecurity incidents in 2023 used phishing as an initial attack vector, vastly surpassing all other methods [2]. As described by the Internet Crime Complaint Center (IC3) unit of the FBI, phishing is "the use of unsolicited email, text messages, and telephone calls purportedly from a legitimate company requesting personal, financial, and/or login credentials" [3]. Further, phishing incidents which were reported to the IC3 were responsible for nearly $18 million in damages [3]. While large, this metric doesn't account for financial damages brought about by the aftermath of phishing emails, such as identity theft, stolen/exfiltrated data, and even the installation of malware on a device. The significant financial losses to both businesses and individuals has prompted research into mitigating the number of successful phishing attacks.

The main ways adopted to reduce the amount of successful phishing attacks are through training user awareness to malicious emails and to adopt a technological approach, where systems actively filter content, looking for suspicious emails or malicious links [4]. Even with the immense energy and research in this area, a recent phishing test report from the Cybersecurity and Infrastructure Security Agency (CISA), revealed that 84% of employees opened a malicious email within 10 minutes of receiving it, clicked on the malicious URL within, then entered some sort of sensitive information into the fake website the URL directed them to [5]. On top of this, it was also determined that 15% of endpoint protection applications and 70% of network security applications, designed to block suspicious traffic, allowed for these malicious documents and links to pass through with being blocked. With a large number of people falling for phishing tactics, and even current technology being fooled, research into successful and accurate classification of phishing links needs to be investigated.

II. URL-based Phishing Detection

Aung and Yamana identify several various strategies commonly used in phishing detection including whitelist solutions, blacklist solutions, content/heuristic approaches, visual similarity comparisons, and URL analysis [4]. In phishing detection, URL analysis involves using different statistical and lexical properties of a given URL to help identify phishing websites [4].

Compared to the other listed solutions, URL analysis remains as one of the stronger strategies in phishing detection due to its advantages. For example, malicious links do not need to be clicked as the analysis happens externally in regards to the website [4]. This provides a significant advantage as many malicious URLs used in phishing attempts can contain malware files, which may begin downloading immediately upon clicking a link. Additionally, URL analysis allows for prevention of zero-day phishing attacks [4]. By looking at the similarities between malicious links, even phishing attempts that have never been seen before can be stopped before they can cause damage.

III. Purpose

With the benefits posed by URL analysis, this paper sets forth to develop a prediction model that can be used to help determine the legitimacy of URLs, aiding the research in phishing detection. This paper investigates various features and models to help determine the best factors in phishing detection. Additionally, it seeks to provide some insight into how a system may perform calculations to derive necessary attributes for analysis, while also functioning in a timely manner.

IV. Methods

The structure of the study can be seen in Figure 1 below. The process involved obtaining 2 different datasets, both from Kaggle [6, 7]. For simplicity, they'll be referred to as Dataset 1 and Dataset 2. Dataset 1 contained 2.5 million instances of URLs with 18 attributes including the URL itself, its label (legitimate or phishing), and several pre-calculated attributes. Its main purpose was to test various attributes related to determining the legitimacy of a URL and create 3 models: a Decision Tree, a Random Forest, and a Naive Bayes model. Dataset 2 contained over 500,000 instances of URLs with only 2 attributes, the URL and its label (good or bad), and was used as a way to see which attributes from Dataset 1 could be obtained in an efficient manner, and also as a way to determine how the first models would hold when tested on new data. The rationale behind calculating the attributes for Dataset 2 was to simulate a real world scenario, where a system will only be given a URL, therefore will need to determine the attributes from it. If attributes can't be gathered in a quick manner, it gives people more time to accidentally click on the link, falling for a phishing attack.

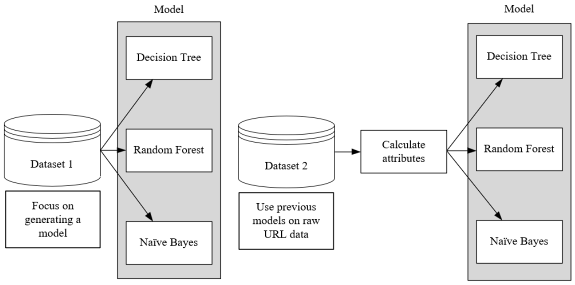


Figure 1: Structure of Study

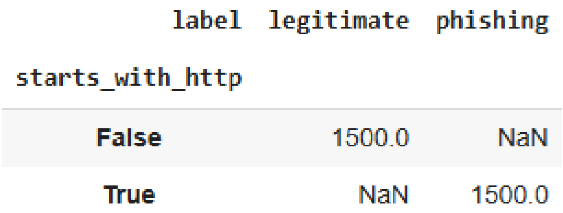
V. Preprocessing of the Initial Dataset

Several issues were found with the initial dataset. Firstly, the original dataset was too large to load for analysis. To solve this issue, 2 random subsets, one containing 1,500 instances of legitimate URLs and the other containing 1,500 instances of phishing URLs, were taken from the original dataset and merged to create one dataset of 3,000 instances. The URLs were kept in equal parts to mimic the original dataset as much as possible.

Once loaded, another issue was quickly found in that the original dataset was removed from Kaggle. This left many questions as to what a few attributes meant, how they were calculated, and more. For the questions that emerged, Aung and Yamana's paper was a great resource in answering them and filling in any blanks [4]. After consulting the paper the attributes in Dataset 1 were defined as follows (attributes highlighted in gray denote that they were later removed for reasons to be explained):

Attribute	Description
url	The name of the URL
source	The database the URL was gathered from
label	Holds "Legitimate" or "Phishing"
url_length	Number of characters in the URL
starts_with_ip	"True" if the URL begins with an IP address
url_entropy	Randomness of characters in URL
has_punycode	"True" if the URL uses punycode
digit_letter_ratio	Ratio of digits to letters
dot_count	Number of "." in URL
at_count	Number of "@" in URL
dash_count	Number of "-" in URL
tld_count	Number of top-level-domains within the URL
domain_has_digits	"True" if the URL contains digits
subdomain_count	Number of subdomains in the URL
nan_char_entropy	Randomness of Non-alphanumeric characters
has_internal_links	"True" if the URL has internal links
who_is_data	Information provided from the "whois" command
domain_age_days	Age, in days, the domain has been running

After describing the attributes, preprocessing of the data began. First, the “starts_with_ip” attribute was modified from its original meaning according to recommendations found within Aung and Yamana’s paper. Instead of just looking at the start of a URL, investigating the entire URL for an IP address can help identify malicious links as attackers may use IP addresses to obscure where a link is directing you [4].



label	legitimate	phishing
starts_with_http		
False	1500.0	NaN
True	NaN	1500.0

Figure 2: Instances starting with http://

Additionally, as shown in Figure 2, it was found that all phishing URLs began with either the “http://” or “https://” protocols while none of the legitimate URLs did. Since only phishing links had these protocols, attributes like “url_length” would be biased to being longer for malicious URLs as all of them would contain several extra characters. Given that it was unknown what protocols the legitimate URLs were using, removing these protocols from the URLs would help remove the bias associated with them. Other attributes affected by resizing the phishing URLs include “nan_char_entropy”, “url_entropy”, and “digit_letter_ratio”. All of these attributes along with the aforementioned “url_length” needed to be adjusted.

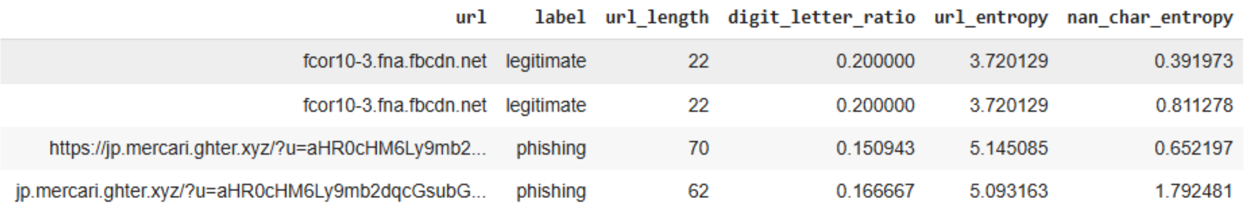
While recalculating “url_length”, “digit_letter_ratio”, and “url_entropy” were simple, however, recalculating “nan_char_entropy” proved to be challenging. When recalculating this attribute, it was found that a different equation was used to initially calculate the entropy than what was being used to recalculate it. While different attribute values were expected to be seen in phishing URLs, different values were also observed in instances which had not been readjusted as seen in Figure 3.

Since the initial dataset was removed from Kaggle, there was no way to go back and reference the original calculation to determine “nan_char_entropy” so, the general equation to calculate the attribute, as highlighted in Aung and Yamana, was used [4].

After this, trimming certain attributes out of the data was conducted. This was done for various reasons such as real world implications, complexity, and more. Real world implications were decided based on what was able and unable to be calculated in Dataset 2 in an efficient manner.

Attribute	Reason for removal
source	In the real world, URLs won’t be coming from a database. Future work to include a database scanner should be looked into
has_punycode	Was unable to determine in an efficient manner
tld_count	Was unable to determine in an efficient manner
domain_has_digits	Redundant with digit_letter_ratio, therefore removed to reduce complexity
has_internal_links	Unable to determine in an efficient manner
who_is_data	Unable to determine in an efficient manner. Also would be redundant with domain_age_days. Further, regulations like GDPR are starting to hide whois data for certain sites, reducing usage
domain_age_days	Unable to determine in an efficient manner

Figure 3: Different nan_char_entropy values



	url	label	url_length	digit_letter_ratio	url_entropy	nan_char_entropy
	fcor10-3.fna.fbcdn.net	legitimate	22	0.200000	3.720129	0.391973
	fcor10-3.fna.fbcdn.net	legitimate	22	0.200000	3.720129	0.811278
	https://jp.mercari.ghter.xyz/?u=aHR0cHM6Ly9mb2...	phishing	70	0.150943	5.145085	0.652197
	jp.mercari.ghter.xyz/?u=aHR0cHM6Ly9mb2dqcGsubG...	phishing	62	0.166667	5.093163	1.792481

VI. Training the Models

After removing the aforementioned attributes, training the models on the dataset ensued. The dataset was split 80/20 into training and testing sets, then fed into the 3 different models. After training the models on Dataset 1, the required attributes were then calculated for Dataset 2 and predictions were made from these attributes using the previously trained models.

The Decision Tree classifier was tuned via adjusting the max depth of the tree. The best performance was found when the max depth was set to 8, attempting to limit overfitting while providing the best performance indicators. For the Random Forest Classifier, the number of trees was adjusted with the best performance found when the model was run over 100 trees. As for the Naive Bayes classifier, there were no parameters to tune.

The accuracy of each model, along with the precision, recall, and F1 measures were recorded. Additionally, confusion matrices for the models were created highlighting the true positive rate (TPR), false positive rate (FPR), true negative rate (TNR), and the false negative rate (FNR).

Figure 4 highlights the results from each model when used on each dataset. In Dataset 1, the Decision Tree and Random Forest classifiers performed about equally, obtaining accuracy scores of 0.88 and 0.87 respectively. However, the Naive Bayes model performed noticeably worse (accuracy of 0.71). When used in Dataset 2, all models experienced worse performance with the Decision Tree and Random Forest classifiers both dropping drastically in accuracy. Interestingly, while the Naive Bayes model also experienced decreased performance, the margin of difference between Dataset 1 and Dataset 2 was significantly smaller than what was observed for the other models (10 points compared to 55). These differences can be seen visually in Figure 5.

Figure 5: Accuracy comparisons

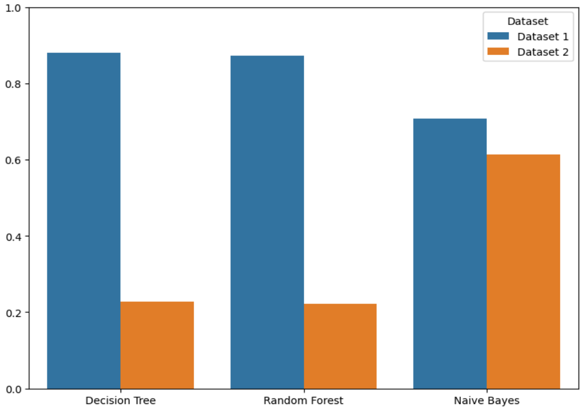
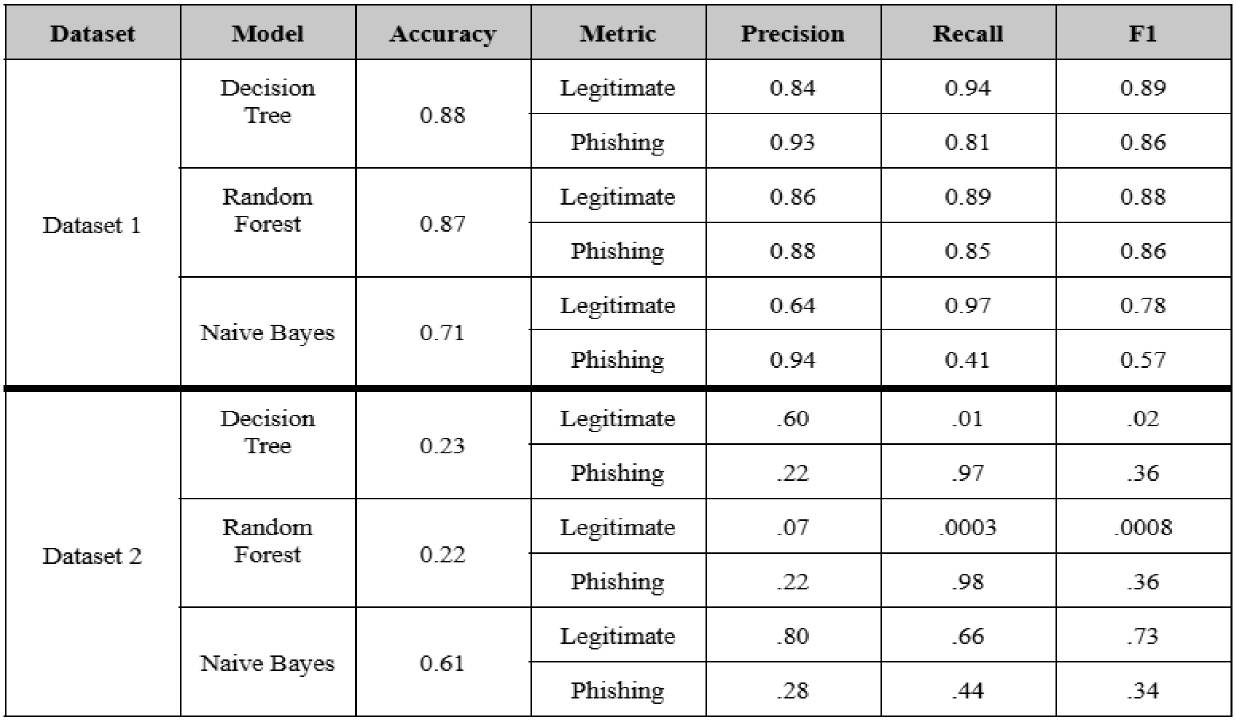


Figure 4: Comparison of results

VII. Model Comparison



Dataset	Model	Accuracy	Metric	Precision	Recall	F1
Dataset 1	Decision Tree	0.88	Legitimate	0.84	0.94	0.89
			Phishing	0.93	0.81	0.86
	Random Forest	0.87	Legitimate	0.86	0.89	0.88
			Phishing	0.88	0.85	0.86
	Naive Bayes	0.71	Legitimate	0.64	0.97	0.78
			Phishing	0.94	0.41	0.57
Dataset 2	Decision Tree	0.23	Legitimate	.60	.01	.02
			Phishing	.22	.97	.36
	Random Forest	0.22	Legitimate	.07	.0003	.0008
			Phishing	.22	.98	.36
	Naive Bayes	0.61	Legitimate	.80	.66	.73
			Phishing	.28	.44	.34

The differences observed between the accuracy measures can be explained in a variety of ways. Given that the two classifiers which required tuning both experienced the largest decrease in accuracy, it is most likely that the specified parameters for the models were overfit for Dataset 1, explaining the huge decrease in accuracy. However, the Naive Bayes model also experienced a drop in accuracy, even though no parameters were tuned in the creation of the classifier. The decreased performance here can likely be explained by the structure of the datasets themselves. Dataset 1 was significantly smaller than Dataset 2 (3,000 instances compared to 500,000). Additionally, Dataset 1 was balanced, containing 1,500 legitimate and 1,500 phishing URLs, while Dataset 2 was imbalanced, containing around 360,000 legitimate URLs and only about 140,000 phishing ones.

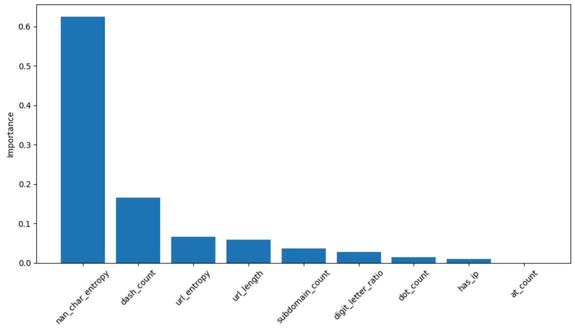
The negative effects of the combination of overfitting and the different structures of the datasets are highlighted within the confusion matrices seen in Figures 6, 7, and 8. The matrices for the models in Dataset 1 help visualize that a majority of URLs were labeled correctly, minimizing the FPR and FNR. Further they demonstrate that in Dataset 2, for the Decision Tree and Random Forest classifiers nearly all instances of URLs were labeled as phishing. And while the Naive Bayes model performed better, it experienced an opposite effect, with a majority of URLs being labeled as legitimate.

The extent to which the confusion matrices lean is exemplified by the measured precision and recall rates of each of the models, as seen in Figure 4. Looking at the Random Forest classifier, the precision values for legitimate and phishing instances of URLs are 0.07 and 0.22 respectively, meaning that the model was correctly predicting only 7% of legitimate URLs and 22% of phishing URLs. The recall values of the model in Dataset 2 were 0.0003 and 0.98, meaning that only 0.03% of all legitimate URLs were even labeled as legitimate but 98% of all phishing emails were labeled as phishing. While the recall metric for phishing URLs looks good, the precision metric helps put it into perspective by demonstrating a low correct prediction rate. Even though nearly all phishing URLs were successfully labeled, the precision metric highlights that a surplus of legitimate URLs were also labeled as phishing, reducing the overall effectiveness of the model.

VIII. Discussion

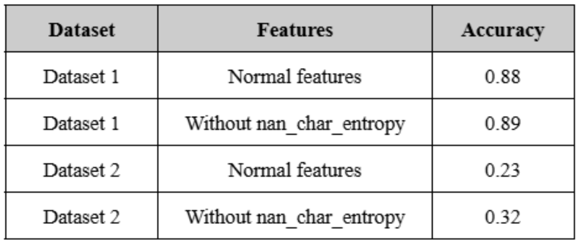
Previous studies have identified that the entropy of non-alphanumeric characters is a good indicator for predicting the legitimacy of URLs [4]. The current work aligns with these sentiments in the fact that the entropy of NaN-characters was the most important feature for making decisions in the Decision Tree classifier as shown in Figure 9.

Figure 9: Feature importance scores



However, when the attribute nan_char_entropy was removed from the dataset, performance was actually shown to increase in both Dataset 1 and Dataset 2. Figure 10 demonstrates the increase by 1 point in Dataset 1 and by nearly 10 points in Dataset 2. These differences can likely be attributed to the recalculation of the nan_char_entropy attribute which occurred while preprocessing the data. Although the equation used in the calculation has been used in other papers [4], the deviation from the original calculation could have established a bais into the dataset.

Figure 10: NaN-character accuracy differences



Dataset	Features	Accuracy
Dataset 1	Normal features	0.88
Dataset 1	Without nan_char_entropy	0.89
Dataset 2	Normal features	0.23
Dataset 2	Without nan_char_entropy	0.32

Another area of interest that needs to be looked into more is the performance of Naive Bayes in both Datasets. While performing worse overall in Dataset 1, the classifier's performance was not drastically reduced in Dataset 2, as was seen in the Decision Tree and Random Forest classifier. This lends way to thinking that a probabilistic approach to URL analysis may be better than a deterministic approach. The Naive Bayes model was able to adapt to new URLs, which is crucial in the real world.

Today, several methods exist to aid cyber criminals in phishing attempts. Tools, such as AI, allow attackers to generate malicious URLs [2]. These tools also allow for attackers to change these bad URLs once they are getting flagged by detection systems at increased rates, helping them to avoid detection and increasing the chance for the attack to be successful [2]. Further, the URLs generated by these modern tools can be made to be extremely similar to legitimate URLs [2]. Between the trademarks of classic phishing URLs and the sophistication of modern ones, malicious links can present themselves in a variety of ways. The constant fluctuation of appearance of these URLs lends to the strengths of probabilistic approaches, which can account for the fact that phishing URLs can present themselves in different ways.

On the other hand deterministic approaches struggle as they rely on sets of rules and patterns, which can easily be manipulated by attackers. For example, various poisoning attacks have been leveraged by attackers to evade detection systems using machine learning [8]. One such attack, data poisoning, involves manipulating the dataset of the system by injecting bad data which can introduce biases and increase the number of incorrect classifications [8]. When attackers alter the training set they can introduce specific biases that conform to their needs. While this attack is detrimental to all machine learning models, the impact on deterministic approaches can be more severe as their rule sets are changed, whereas with probabilistic approaches, their nature allows them more flexibility. Additionally, by intentionally crafting malicious traffic, attackers can conduct false positive poisoning, an attack method that introduces a large number of false positive alerts, distracting the system and the security team from real threats [8]. For URLs this can take the form of an attacker using the aforementioned tools to generate fake malicious links, flooding the system with threat flags, then using the tools again to craft a well-made phishing URL that is able to sneak by the system and the security team. For deterministic approaches, due to their pattern detection may allow for attackers to easily manipulate them to generate false positive results whereas probabilistic approaches have more flexibility in their decision making.

IX. Limitations and Future Research

Several areas of the study could be improved. For example, several predictors, like domain_age_days, were left out of the model. While methods exist to determine these, the size of Dataset 2 was too large to do them efficiently. For example, it is possible to determine the age of a domain in a URL via the whois command. However, trying to determine the age of a domain for over 500,000 URLs took well over 20 minutes. As this study attempted to mimic real world systems calculating attributes in real time, this time was determined to be too long and was therefore scratched from analysis. Future work should look into determining a more efficient way of analysis.

Another area of limitation was in the removal of Dataset 1 from Kaggle. Other than the previous issues mentioned before, no reason was given for the removal of the dataset which can lead to concerns about its validity and the ethical collection of data. For example, a reason the dataset could have been removed was because it contained biased data. While the bias of phishing URLs containing their protocols but legitimate URLs not, was addressed, perhaps additional biases went uncaught while preprocessing the data.

X. Conclusion

Phishing detection has become a crucial area of cybersecurity, playing a vital role in protecting individuals and organizations from financial loss, data breaches, and reputational damage. This study brings to light the potential benefits of URL-based analysis as an effective tool for identifying malicious links. By comparing deterministic models like Decision Tree and Random Forest with the probabilistic Naive Bayes classifier, a potential strength of probabilistic approaches was revealed for URL-based analysis. Limitations were also highlighted, such as the restriction of certain attributes for real-time application and potential biases in the datasets used. Future work should focus on developing more efficient methods for feature extraction and potentially even exploring hybrid approaches that combine deterministic precision with probabilistic flexibility. Advancing phishing detection systems is crucial to staying ahead of evolving threats, ensuring a safer digital environment for all users.

References

[1] B. B. Gupta, N. A. G. Arachchilage, and K. E. Psannis, “Defending against Phishing Attacks: Taxonomy of Methods, Current Issues and Future Directions,” 2017. https://arxiv.org/abs/1705.09819

[2] H. Ozsahan and D. Worthington, “50+ Phishing Attack Statistics for 2024,” *JumpCloud*, Mar. 12, 2024. https://jumpcloud.com/blog/phishing-attack-statistics

[3] Internet Crime Complaint Center, “Internet Crime Report,” 2023. Available: https://www.ic3.gov/AnnualReport/Reports/2023_IC3Report.pdf

[4] E. S. Aung and H. Yamana, “URL-based Phishing Detection using the Entropy of Non-Alphanumeric Characters,” *Proceedings of the 21st International Conference on Information Integration and Web-based Applications & Services*, Dec. 2019, doi: https://doi.org/10.1145/3366030.3366064.

[5] Cybersecurity & Infrastructure Security Agency, “Phishing,” 2023. Available: https://www.cisa.gov/sites/default/files/2023-02/phishing-infographic-508c.pdf

[6] S. Agudelo, “Phishing URL Detection,” *Kaggle*, 2024. https://www.kaggle.com/datasets/sergioagudelo/phishing-url-detection

[7] T. Tiwari, “Phishing Site URLs,” *Kaggle*, 2020. https://www.kaggle.com/datasets/taruntiwarihp/phishing-site-urls

[8] CrowdStrike, “What Is Data Poisoning?,” 2024. https://www.crowdstrike.com/en-us/cybersecurity-101/cyberattacks/data-poisoning/

Figure 6: Decision Tree Confusion Matrices

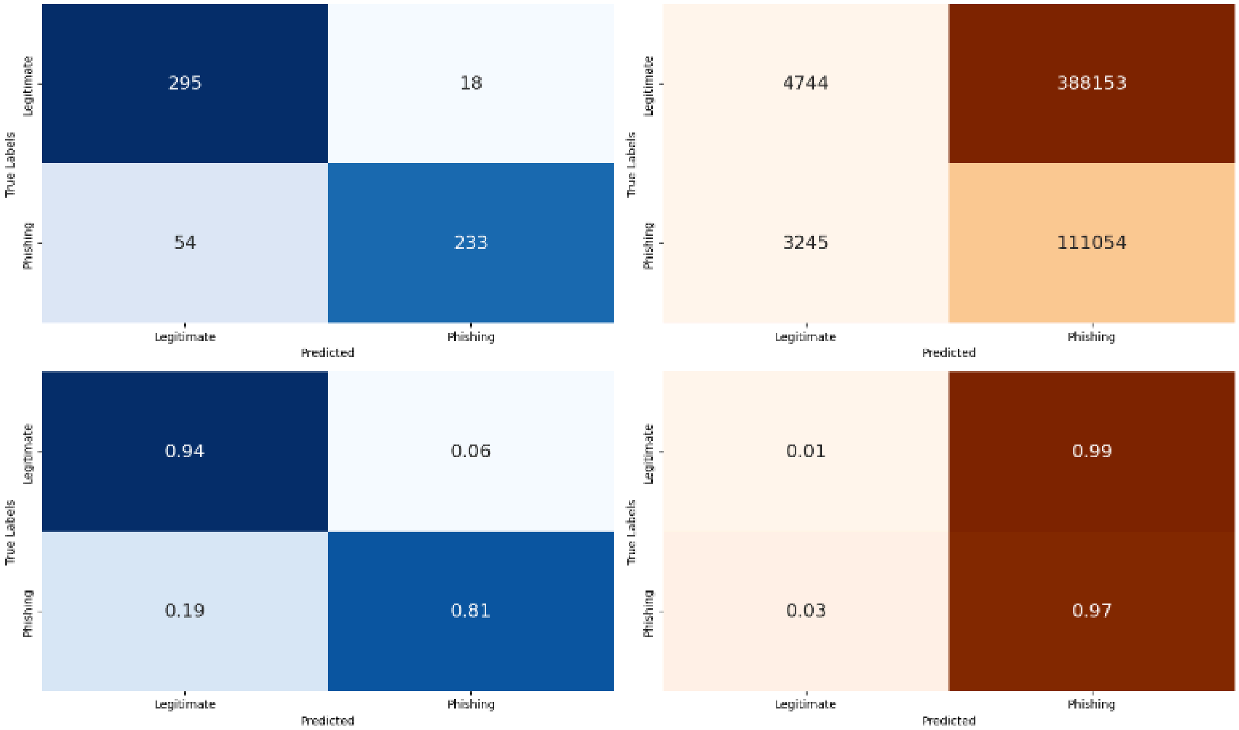


Figure 7: Random Forest Confusion Matrices

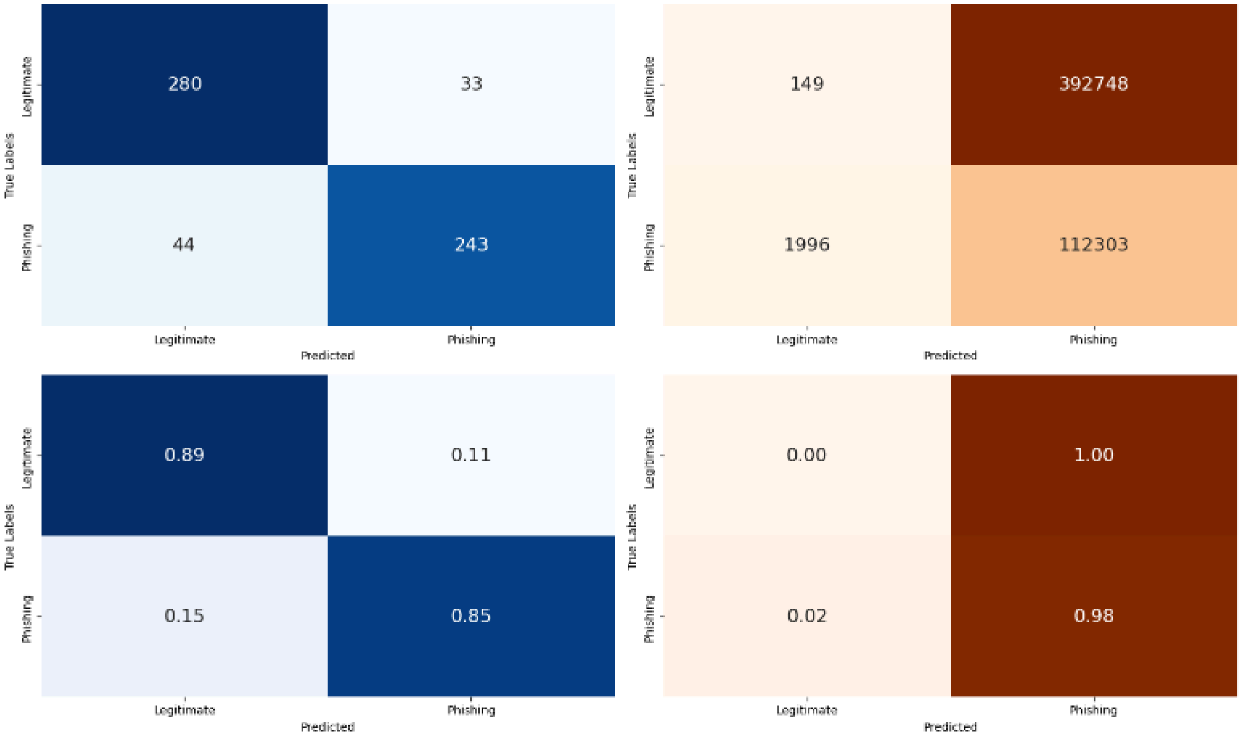
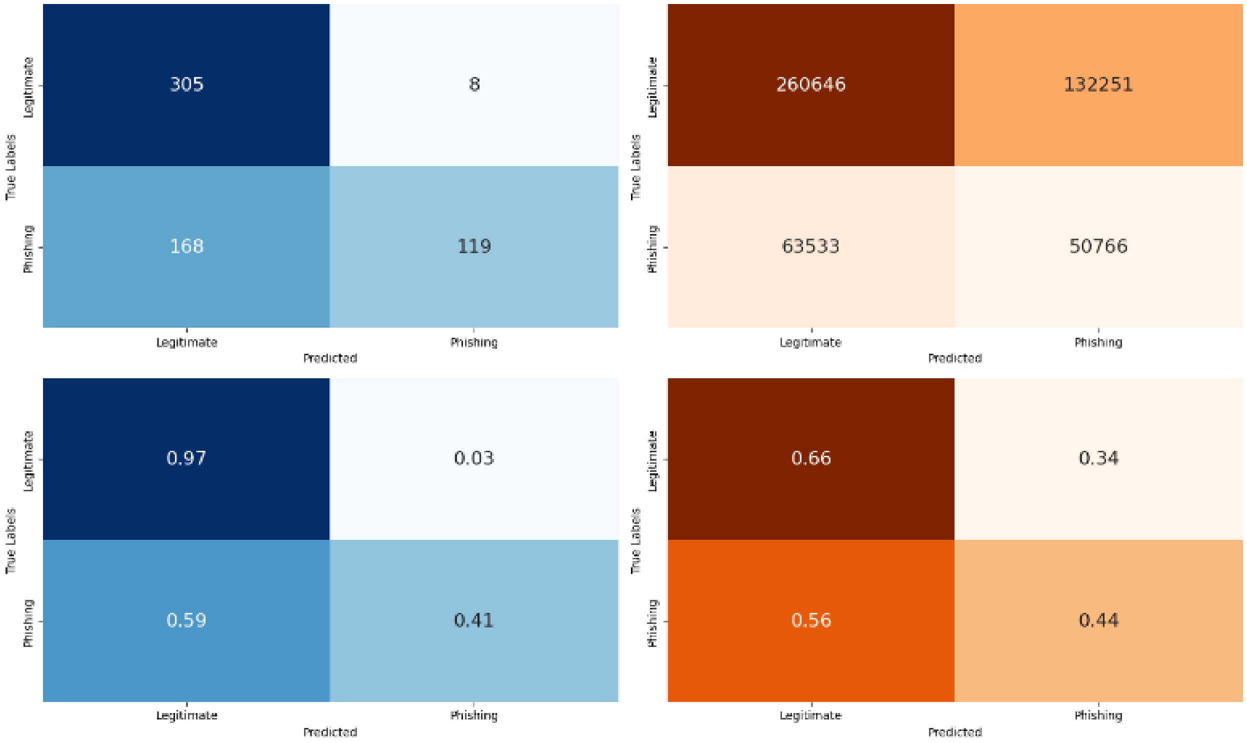


Figure 8: Naive Bayes Confusion Matrices


True Labels
Legitimate
Phishing
Legitimate
Phishing
Predicted
True Labels
Legitimate
Phishing
Legitimate
Phishing
Predicted
True Labels
Legitimate
Phishing
Legitimate
Phishing
Predicted
True Labels
Legitimate
Phishing
Legitimate
Phishing
Predicted