Replication and Extension of Advanced Phishing Detection:

Integrating Machine Learning with Email Services for Malicious URL Identification

Tyler Beasley, Keshawn Blakely, Matthew Tieman

University of South Carolina

CSCE 585

Abstract

In this digital age, Cybercrime is constantly increasing along with technology. This puts personal information, security, and valuable assets at risk, highlighting the importance of reliable protection. Among these crimes, phishing attacks are known to be the most prominent. This introduces our objective of detecting phishing attempts through tokenization, a process used by machine learning models to analyze and classify the given URL. Our work replicates and extends on the research paper, "Multimodal Phishing URL Detection Using LSTM, Bidirectional LSTM, and GRU models" with a focus on tokenization and integration with email services. Our approach involved testing all three recurrent neural network models (LSTM, Bidirectional LSTM, and GRU) and comparing the performances with real-time detection.

Introduction

As technology provides the backbone to many aspects of today's society, cybercrimes could potentially lead to catastrophic events, drastically impacting the way we live our daily lives. For instance, in 2023, the Internet Crime Complaint Center (IC3) received over 880,000 complaints and reported over $12 billion in losses with a 22% increase from 2022 [1]. Current trends show that phishing takes the lead among these various cybercrimes. A phishing attack is a method used by an attacker where they pretend

to be a legitimate entity such as a website or business to obtain otherwise secure personal information. These attacks may come in a variety of forms, making it difficult for many people to distinguish a potential attack. Some popular methods involve malicious email attachments, fake apps/websites, and even fraudulent QR codes. Due to their simplicity, a malicious URL is the primary method used by these attackers. Highlighting this concern, between January 2022 and December 2023, there has been an increase of 8 million phishing attacks through email services [2]. As these attacks evolve, there needs to be a corresponding solution to protect against them. This constant evolution of attacks makes it difficult to counter with static solutions such as blacklisting a URL. This introduces the need for dynamic approaches like machine learning. To address this issue Recurrent Neural Networks (RNN) are designed to process sequential data with the ability to recall information from earlier in the sequence making it ideal for detecting malicious URLs. The specific model used in our extension is the Gated Recurrent Unit (GRU) which is a more advanced version of traditional RNNs that uses gating mechanisms to control which information is kept or discarded.

Related Work

Our project is derived from the published work, “Multimodal Phishing URL Detection Using LSTM, Bidirectional LSTM, and GRU Models” [3] where the authors explore three RNN models: LSTM, Bidirectional LSTM, and GRU. Their work advocates for the use of deep learning to detect malicious URLs, focusing on tokenization and model evaluation. Our project not only replicates these models but also extends their application by integrating them into a system for real-time detection with email services. Our work evaluates the performance of all three models, replicating the training and validation accuracy results. However, rather than employing all three models, our project focuses exclusively on the GRU model, which has proven to offer the best balance between speed and accuracy.

Data

The data used for our model was sourced from PhishTank, a publicly available database that comprises of over 450,000 URL addresses labeled as malicious or benign. To use this data, it required preprocessing, allowing it to be understood by the model. First, the data was loaded into a pandas data frame with a URL and label columns. These URLs are then tokenized and converted into numerical sequences, ensuring that the data can be understood by an RNN-based architecture. Next, to ensure the data is of uniform length, the tokenized URL sequences are padded to 100 characters. This padding allows the model to handle varying lengths efficiently. Lastly, to finish preparation, the labels were converted to a binary 1 or 0, indicating malicious or benign. After the data has been properly preprocessed it can then be split (80-20) for train-test split validation.

Methods

Our approach to developing machine learning based phishing detection leverages RNNs to detect malicious URLs despite their variability and complexity. Alternative to static methods such as blacklisting, rule-based systems, and regular expression matching, this is an efficient approach due to the ability of RNN models to effectively process sequential data. By analyzing the sequence of characters or tokens in a URL, RNNs can identify trends and features that would otherwise be missed by static phishing detection, enabling it to keep up with evolving tactics used by modern phishing attacks. In addition, we explored the comparison between our deep learning RNN based model and traditional machine learning using Random Forest classifier—a non-LLM model that builds decision trees to make predictions based on predefined features. Random Forest was selected because it is widely used in cybersecurity applications for its ease of implementation. When tested on our data set Random Forest performed poorly, indicated by the results below in Figure 1: Test Results of Random Forest

```
Accuracy: 0.6459527300191035
Classification Report:
              precision    recall  f1-score   support

           0       0.84      0.66      0.74     68921
           1       0.35      0.58      0.44     21115

    accuracy                           0.65     90036
   macro avg       0.59      0.62      0.59     90036
weighted avg       0.72      0.65      0.67     90036
```

Figure 1: Test Results of Random Forest

As seen above, the non-LLM model achieved an accuracy of 64.6% which would be sufficient for older phishing techniques. On the other hand, the three RNN models we tested achieved much higher accuracy percentages displayed in Figure 2.

Model	Accuracy
Long Short-Term Memory (LSTM)	99.45%
Bidirectional Long Short-Term Memory (Bi-LSTM)	99.02%
Gated Recurrent Unit (GRU)	99.60%

Figure 2: RNN Model Accuracy

The three RNN models performed very well in our testing of the whole dataset with all test accuracies being 99%. However, the differences in the performance of these models occur within the latency and strain on the hardware. To decide which model would be best for real-time detection, we will combine accuracy testing results with load-testing simulations to evaluate how your phishing detection models handled concurrent requests in a real-world-like scenario.

Experiments

Before beginning the experiment, we needed a process to gather the data and train the model. The high-level process of this experiment is as follows:

- Gather the phishing data that comprises of web searches, enterprise data, and other sources

- Perform feature extraction and preprocess the data
- Utilize the data to train the Deep learning Classifiers
- Save each trained model
- Integrate the model into a web application
- Perform load-bearing tests

This process is illustrated in the diagram below:

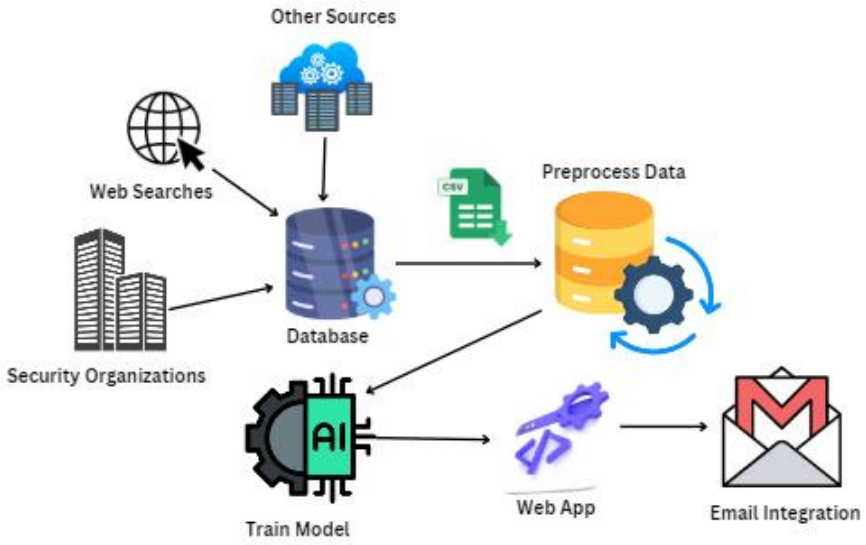


Figure 3: Workflow Diagram

The data set used for this project contained over 450,000 links gathered from PhishTank. The bar graph below in Figure 4 shows the distribution of benign and phishing links.

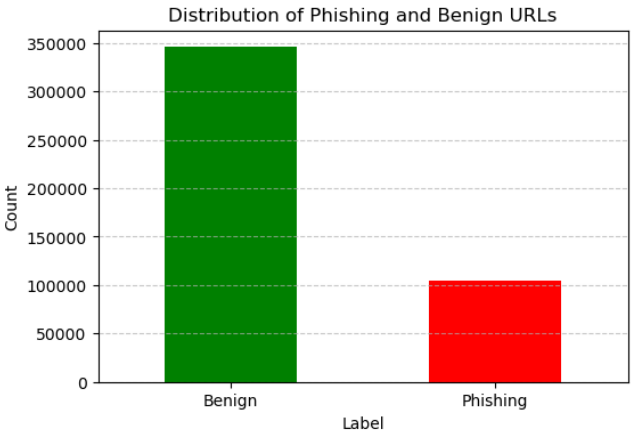


Figure 4: Link Distribution

Each model was tested with Each model was tested with the data set which went through a tokenization process, preparing it for the RNN-based architectures. This involved breaking the URL down into numerical sequences which were then padded to 100 characters to ensure they met the input requirements. Each model's performance was graphed in a confusion matrix, displaying the accuracies of overall predictions. The LSTM model which was tested to be 99.45% had 0.35% false positive and 0.19% false negative scores in Figure 5. The Bi-LSTM model which was tested to be 99.02% had 0.32 % false positive and 0.36 % false negative scores in Figure 6. Lastly, The GRU model which was tested to be 99.60% had 0.26 % false positive and 0.12 % false negative scores in Figure 7. Additionally, each model was tested to create a training and validation accuracy chart. The charts all replicated from the research paper and have shown a high resemblance seen in figures 8-10. However, we used a lower number of

epochs as we are working with much slower hardware. Nevertheless, the general trends remained the same for the first 10 epochs.

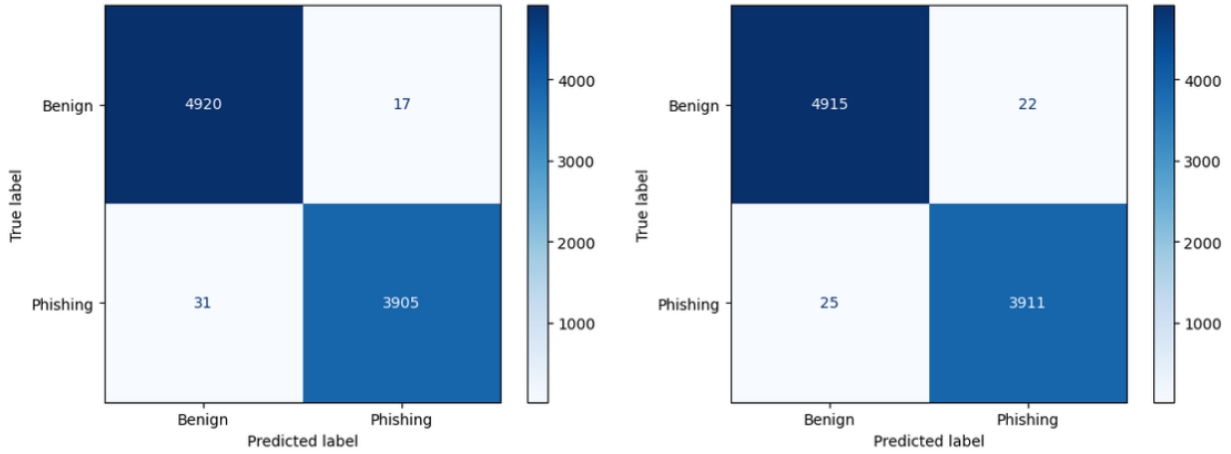


Figure 5-6: LSTM(left) and BiLSTM(right) Confusion Matrix

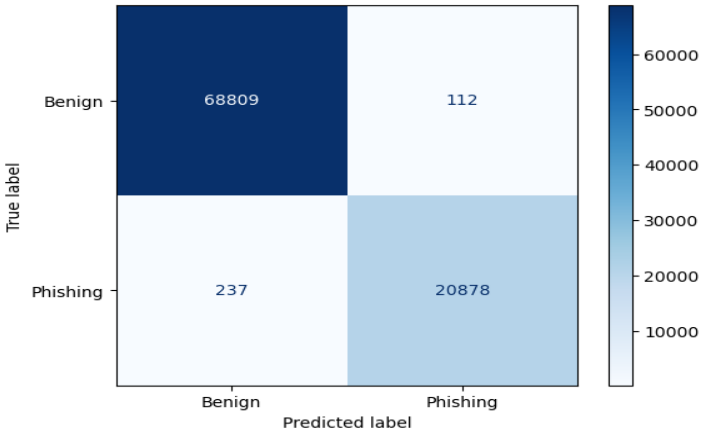


Figure 6: GRU Confusion Matrix

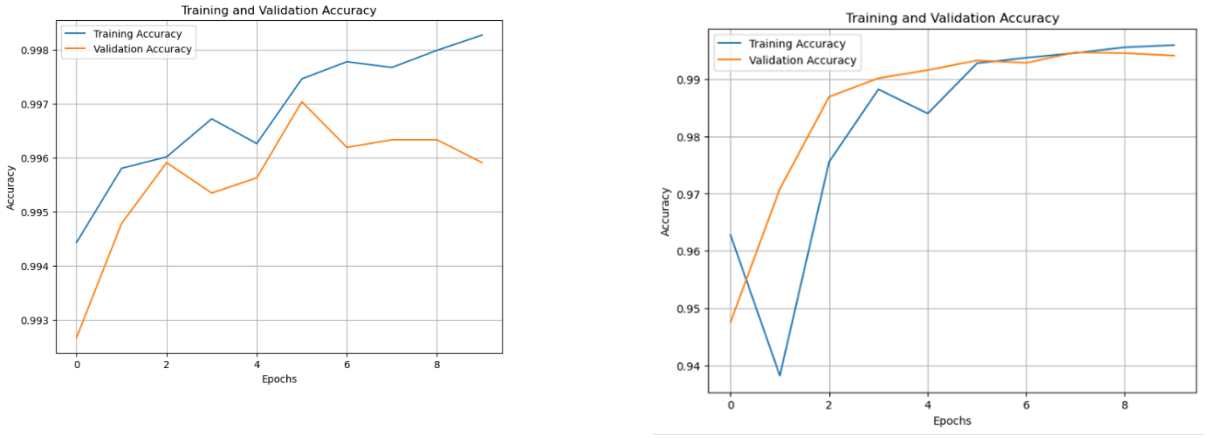


Figure 7-9: LSTM(left) and BiLSTM (right) Training and Validation Accuracy

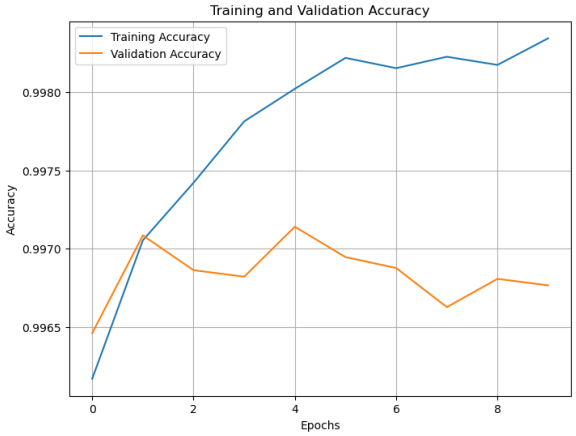


Figure 8: GRU Training and Validation Accuracy

Overall, every model achieved sufficient results and the testing and validation accuracies with a low percentage of false positives and false negatives which gave reason to test each model when integrated with a web-based application. Next, the models are given load-bearing tests using locust, to simulate a medium load of 100 users for 30 minutes and a heavier load of 500 users for 10 minutes.

Model	Requests	Failures	Median Response Time (ms)	Avg Response Time(ms)	Min Response Time (ms)	Max Response Time (ms)	Requests/s	Failures/s
GRU	35175	0	2100	2101.109	2051.363	2392.234	19.54203	0
LSTM	35233	0	2100	2105.836	2051.345	2360.184	19.57381	0
BiLSTM	35117	0	2100	2113.228	2051.292	2431.071	19.51131	0

Model	Requests	Failures	Median Response Time (ms)	Avg Response Time(ms)	Min Response Time (ms)	Max Response Time (ms)	Requests/s	Failures/s
GRU	34289	12577	5900	5702.544	3922.1665	9499.34	57.134340	20.956534
LSTM	34315	12845	6000	5698.6817	3357.4351	8613.556	57.174445	21.401887
BiLSTM	33816	12473	6000	5811.4993	4064.4171	13439.819	56.344	20.7825

Figure 11-12: Load bearing test results for medium load (top) and heavy load (bottom)

These tests were inhibited by our testing environment, as we had to perform them locally on a CPU, which limits our testing abilities as compared to using a GPU based device. Still, it can be seen that the models performed similarly under medium load, with the GRU filling requests slightly quicker than the other models. In the heavier tests, however, the GRU based model seemed to handle the large loads better, having a lower median response time than the other models, and having less failures. However, the bulk of the failures can be attributed to the testing environment not being able to handle the users, rather than the models themselves. We decided that the GRU based model was the most sufficient for real-time application. Lastly, using Google API, the model was successfully integrated with Gmail where it labels the full stream of user emails as phishing or benign.

Conclusion

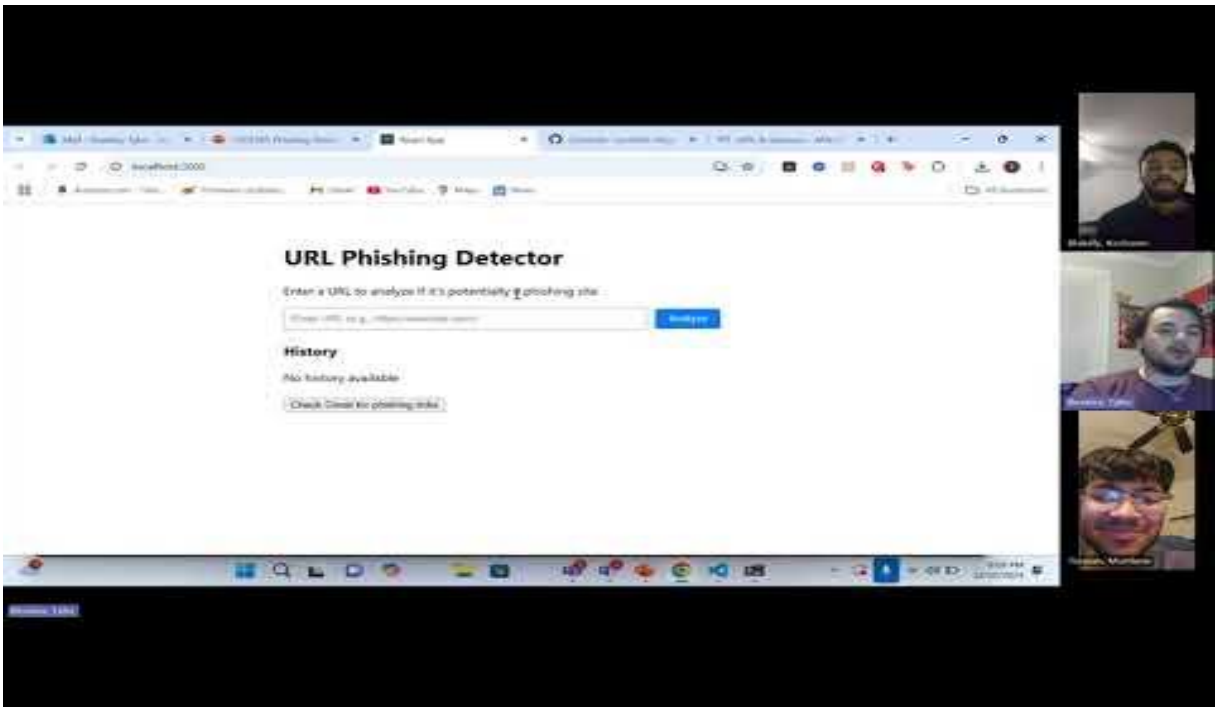
In conclusion, the integration with Gmail, utilizing Google API successfully demonstrated the model's ability to classify email streams in real time, highlighting the practical feasibility of our approach. The key results are that all models are capable for real-time malicious URL detection in email services, but the GRU model specifically provides an excellent balance of computation efficiency and accuracy. While the Bi-LSTM has the best accuracy, its downside is in its computational intensity caused by its bidirectional

architecture. Future directions for this project include deploying the model in a browser extension and adding security features that flags/blocks any malicious URLs embedded in emails. Additionally, feature fusion could enhance detection by integrating URL-based features with email metadata such as sender domain and content patterns.

References

1. FBI Internet Crime Complaint Center. (2023). 2023 Internet Crime Report. Retrieved from https://www.fbi.gov/services/cjis/ic3
2. Petrosyan, A. (2024, September 23). *Global number of e-mail phishing attacks 2022-2023*. Statista. Retrieved from https://www.statista.com
3. Roy, S.S.; Awad, A.I.; Amare, L.A.; Erkihun, M.T.; Anas, M. Multimodel Phishing URL Detection Using LSTM, Bidirectional LSTM, and GRU Models. Future Internet 2022, 14, 340. https://doi.org/10.3390/fi14110340

Video Recording: https://youtu.be/61ZVA_9x3Vo



GitHub: https://github.com/csce585-mlsystems/Phishing-Detection

Slides: https://github.com/csce585-mlsystems/Phishing-Detection/blob/024c058f38acea93f8f240e55e245741de2919a8/CSCE585%20Phishing%20Detection%20Final.pdf