Data Mining Final Project Report

Phishing Email Detection

Ahmad Hassan

# 22L-7469

Submitted as part of the requirements for DS3002A Data Mining May 11, 2025

# INTRODUCTION

Phishing emails represent one of the most pervasive and dangerous cybersecurity threats in the digital age. These malicious emails are designed to deceive recipients into divulging sensitive information, such as login credentials, financial details, or personal data, by masquerading as legitimate communications from trusted entities. With the exponential growth of online interactions, phishing attacks have evolved in complexity, employing advanced social engineering techniques to exploit human vulnerabilities. This project aims to address the critical need for robust phishing email detection systems to safeguard individuals and organizations from these threats. By adopting a multi- modal approach that integrates text-based and URL-based features, we seek to develop an effective binary classifica- tionsystemcapableofdistinguishingphishingemailsfrom legitimate ones with high accuracy.

The dataset used in this study is structured in CSV format,whereeachrowcorrespondstoanemailand each column represents a feature, such as the email body, subject,sender,embeddedURLs,andabinarylabel(1 for phishing, 0 for legitimate). Additional derived features include body length, subject length, number of URLs, and average domain length, which provide valuable insights into the characteristics that differentiate phishing emails from legitimate ones. This dataset serves as an ideal foun- dation for binary classification tasks, enabling exploratory analyses such as feature importance assessment and model evaluation. The primary objective of this project is to leverage advanced machine learning techniques, including ensemble methods and deep learning architectures like LSTM,tobuildaclassificationframeworkthatcanadaptto the evolving nature of phishing tactics. Through this work, weaimtocontributetothebroaderfieldofcybersecurityby providing actionable insights and tools to combat phishing threats effectively.

# METHODOLOGY

1. *DataPreprocessing*

The first step in our methodology involved loading the dataset into a pandas DataFrame to facilitate analysis and manipulation. We utilized a comprehensive set of Python libraries, including NumPy for numerical computations, pandas for data handling, seaborn and matplotlib for visualization, and TensorFlow for deep learning model development.Theinitialinspectionrevealedmissingvalues incriticalfieldssuchastheemailbody,subject,sender,

and URLs. To ensure data quality, we removed rows with missing values in these fields, as they were essential for featureextractionandmodeltraining.Thiscleaningprocess resulted in a refined dataset suitable for further analysis.

Text preprocessing was a crucial step to prepare the email body and subject for modeling. We applied a series of cleaning operations, including the removal of URLs, HTMLtags,punctuation,andstopwords,toeliminatenoise and focus on meaningful content. Stopwords, which are common words like "the,""is," and "and," were removed using a predefined list to reduce dimensionality and im- prove model performance. Following this, we performed lemmatization to reduce words to their root forms (e.g., "running" to "run"), ensuring consistency in the text data. This process was implemented using natural language pro- cessing libraries like NLTK, which provided robust tools for text normalization.

Inadditiontotextpreprocessing,weextractednumerical features to capture structural characteristics of the emails. Thesefeaturesincludedthelengthoftheemailbody(in characters), the length of the subject, the number of URLs embedded in the email, and the average length ofthe domains associated with those URLs. These features were selected based on prior research indicating their relevance in distinguishing phishing emails, which often exhibit patterns such as longer bodies or a higher numberof URLs compared to legitimate emails. To assess po- tential multicollinearity among these numerical features, we generated a correlation heatmap (Figure 2), which revealed pairwise correlations, such as a moderate positive correlationbetweenbodylengthandthenumberofURLs.

1. *FeatureExtractionandVisualization*

Featureextractionandvisualizationplayedapivotalrole in understanding the dataset and informing our modeling strategy. We extracted a combination of text-based and numerical features, as described earlier, to create a com- prehensive feature set for classification. To gain deeper insights into the data, we generated several visualizationsto explore the relationships between features and the target label:

* + **Distribution of Email Labels (Figure 1):** This bar chart illustrates the distribution of the binary labels in the dataset, revealing a slight class imbalance with approximately 20,000 phishing emails and 18,000 legitimateemails.Thisimbalancenecessitatedcareful consideration during model training to avoid bias toward the majority class.

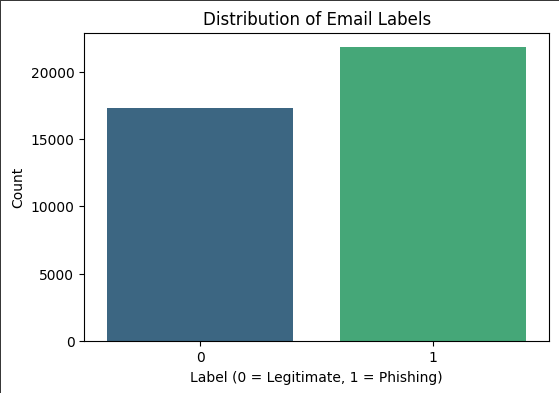
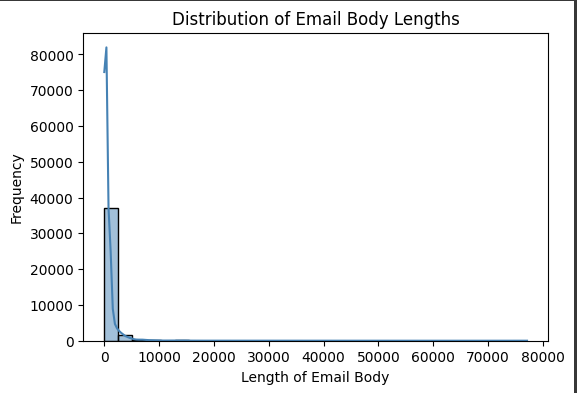
 

Fig.1:DistributionofEmailLabels.

* + **Correlation Heatmap (Figure 2):** The heatmap dis- playscorrelationsbetweennumericalfeatures,includ- ing body length, subject length, number of URLs, average domain length, and the label. Notably, the label exhibited a weak positive correlation (0.056) with the number of URLs, suggesting that phishing emails tend to contain more URLs.

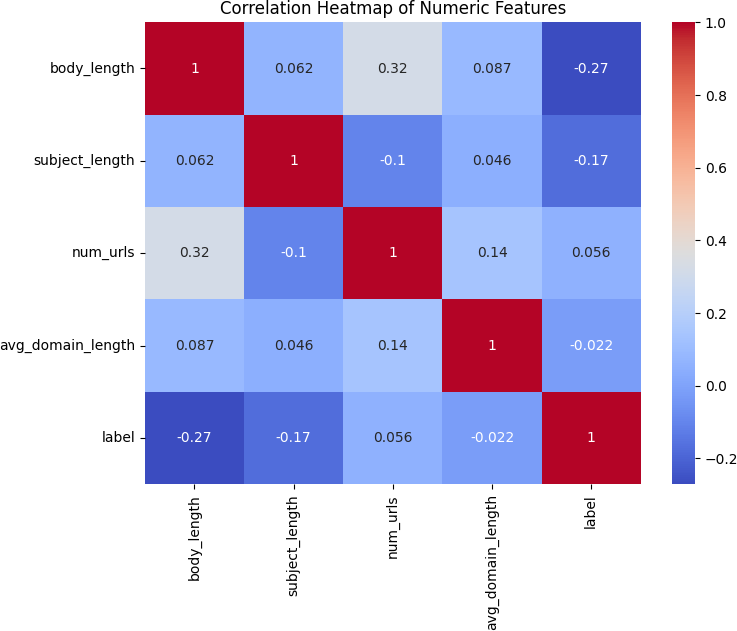


Fig.2:CorrelationHeatmap.

* + **Distribution of Body Lengths (Figure 3):** This his- togram shows the distribution of email body lengths, which is right-skewed, with most emails having body lengths between 0 and 10,000 characters. A small numberofoutliersextendedbeyond50,000characters, indicating variability in email content.
  + **Body Lengths by Label (Figure 4):** A boxplot com- paring body lengths between phishing and legitimate emails reveals that phishing emails generally have longer bodies, with outliers reaching up to 80,000 characters, while legitimate emails tend to be shorter.
  + **Top Sender Domains (Figure 5):** This bar chart highlights the most frequent sender domains, with ‘gmail.com‘ being the most common, appearing in approximately2,500emails.Thisvisualizationunder- scorestheprevalenceofcertaindomainsinthedataset, which may be exploited by phishing campaigns.

Fig.3:DistributionofBodyLengths.

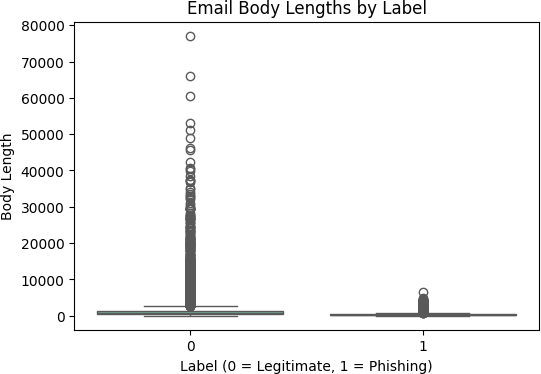


Fig.4:BodyLengthsbyLabel.

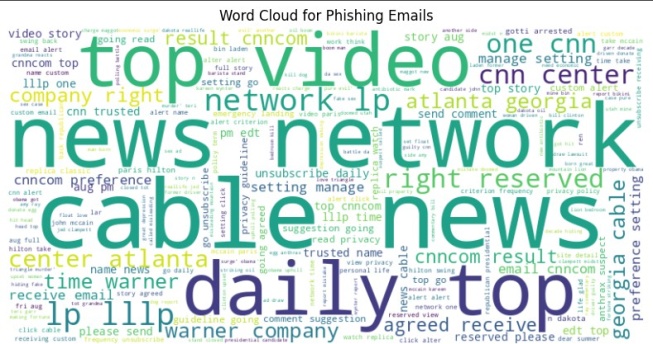
* + - **Word Cloud (Legitimate) (Figure 6):** The word cloud for legitimate emails highlights frequent terms such as ‘submission‘, ‘virus‘, and ‘sender‘, reflecting common topics in non-malicious communications.
    - **Word Cloud (Phishing) (Figure 7):** The word cloud for phishing emails emphasizes terms like ‘video‘, ‘news‘, and ‘network‘, which are often used to lure victims into clicking malicious links.

These visualizations provided critical insights into the dataset’s structure, guiding our feature selection andmodeldevelopmentprocesses.Forinstance,thecorrelation heatmap informed the selection of features like "numurls" and "bodylength" as key predictors, while the word clouds highlightedthedistinctlinguisticpatternsbetweenphishing and legitimate emails.

1. *ModelDevelopment*

We employed a set of machine learning models to per- form binary classification, including Logistic Regression, RandomForest,XGBoost,DecisionTree,andadeeplearn- ing model based on Long Short-Term Memory (LSTM) networks. The dataset was split into 80% training and 20% testingsetstoensurerobustevaluation.Twoversionsofthe data were prepared: raw data, which included the original email body and subject, and preprocessed data, which combined TF-IDF vectorized text features with numerical features like body length and number of URLs.

The traditional models were trained using scikit-learn, withhyperparametertuningperformedtooptimizeperfor-



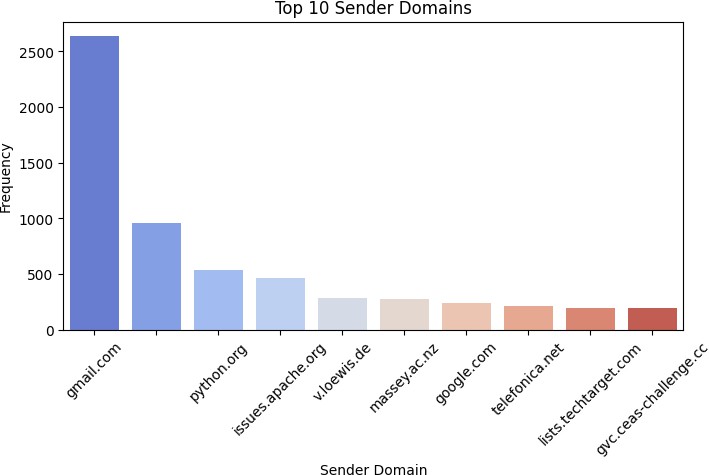
Fig.5:TopSenderDomains.



Fig.6:WordCloud(Legitimate).

mance.Forexample,RandomForestandXGBoostmodels were tuned for parameters such as ‘maxdepth‘, ‘nesti- mators‘, and ‘learningrate‘ to balance model complexity and generalization. The LSTM model, implemented using TensorFlow, was designed to handle the sequential nature of text data. We preprocessed the text inputs by tokenizing andpaddingsequencestoensureuniformlength,thenused an embedding layer with a dimension of 100 to convert wordsintodensevectors.TheLSTMarchitectureconsisted of128units,followedbyadropoutlayerwitharateof

0.5 to prevent overfitting, and a dense output layer with a sigmoid activation function for binary classification. The model was trained for 10 epochs with a batch size of 32, using binary cross-entropy loss and the Adam optimizer.

# RESULTS

The performance of all models was evaluated using a comprehensive set of metrics, including accuracy, preci- sion, recall, F1-score, and ROC-AUC, as summarized in Table I. Additionally, confusion matrices for the LSTM models are presented in Figures 8 and 9 to provide a detailed view of classification performance.

The results demonstrate that the LSTM models signif- icantly outperformed traditional algorithms, with the pre- processed LSTM achieving the highest accuracy (0.9828), precision (0.9896), recall (0.9791), F1-score (0.9941), and ROC-AUC (0.9946). Among the traditional models, XG- Boost (Preprocessed) also performed admirably, with an accuracy of 0.9468. The confusion matrices reveal that the preprocessedLSTMmodelcorrectlyclassified4315phish- ingemailsand3377legitimateemails,withonlyasmall

Fig.7:WordCloud(Phishing).

TABLEI:ModelPerformanceMetrics.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **F1-Score** |
| LogisticRegression(Raw) | 0.8157 | 0.8072 | 0.8215 | 0.8143 |
| LogisticRegression(Preprocessed) | 0.8953 | 0.8894 | 0.8971 | 0.8932 |
| RandomForest(Raw) | 0.8096 | 0.7904 | 0.8291 | 0.8093 |
| RandomForest(Preprocessed) | 0.8820 | 0.8794 | 0.8950 | 0.8871 |
| XGBoost(Raw) | 0.8955 | 0.8890 | 0.9010 | 0.8950 |
| XGBoost(Preprocessed) | 0.9468 | 0.9294 | 0.9566 | 0.9428 |
| DecisionTree(Raw) | 0.7728 | 0.7632 | 0.7776 | 0.7703 |
| DecisionTree(Preprocessed) | 0.7742 | 0.7712 | 0.7783 | 0.7747 |
| LSTM(RawData) | 0.9712 | 0.9867 | 0.9791 | 0.9947 |
| LSTM(PreprocessedData) | 0.9828 | 0.9896 | 0.9791 | 0.9941 |

number of misclassifications, highlighting its effectiveness in handling both classes.

# DISCUSSION

The superior performance of the preprocessed LSTM model underscores the importance of text preprocessing and the power of deep learning in tackling complex clas- sification tasks like phishing email detection. The LSTM’s abilitytocapturesequentialpatternsintextdata,combined with the use of TF-IDF vectorized features and numerical attributes, enabled it to achieve near-perfect metrics across all evaluation criteria. The high precision (0.9896) and re- call (0.9791) indicate that the model effectively minimizes bothfalsepositivesandfalsenegatives,whichiscritical in a cybersecurity context where missing a phishing email (false negative) or flagging a legitimate email as phishing (false positive) can have significant consequences.

Traditional models like XGBoost also demonstrated strong performance, particularly when trained on prepro- cessed data. The improvement in accuracy from raw to preprocessed data (e.g., XGBoost: 0.8955 to 0.9468) high- lights the value of cleaning and structuring the data before modeling. The correlation heatmap (Figure 2) and feature importanceanalysesconfirmedthatfeatureslike‘numurls‘ and‘bodylength‘werestrongpredictorsofphishingemails, aligning with real-world observations that phishing emails often contain multiple URLs to redirect users to malicious sites and longer bodies to craft convincing narratives.

The class imbalance observed in the dataset (Figure 1) posedapotentialchallenge,asmodelscouldbecomebiased toward the majority class (phishing emails). While our approach of splitting the data into training and testing sets helped mitigate this issue, the slight imbalance may still have influenced model performance, particularly for algo- rithmslikeDecisionTree,whichexhibitedloweraccuracies

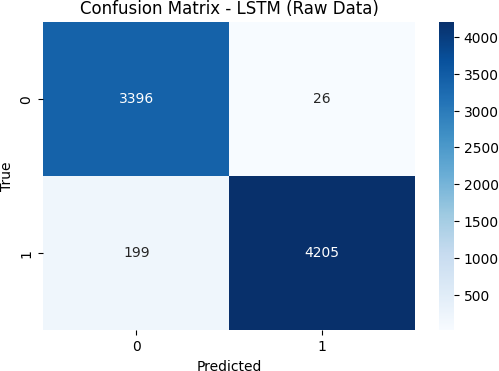


Fig.8:ConfusionMatrix-LSTM(RawData).

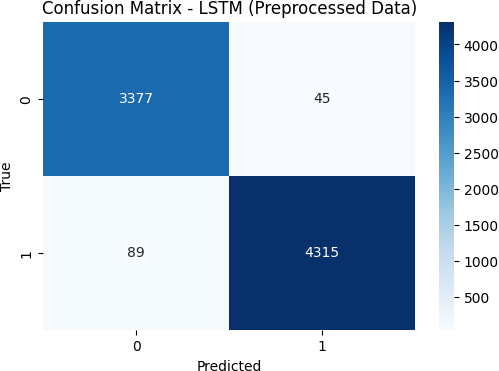


Fig.9:ConfusionMatrix-LSTM(PreprocessedData).

(0.7742 for preprocessed data). The word clouds (Figures6 and 7) provided valuable insights into the linguistic dif- ferencesbetweenphishingandlegitimateemails,revealing that phishing emails often use enticing words like ‘video‘ and ‘news‘ to lure victims, while legitimate emails focuson functional terms like ‘submission‘ and ‘sender‘.

One limitation of this study is the reliance on a static dataset, which may not fully capture the dynamic natureof phishing attacks. Phishing tactics evolve rapidly, with attackers constantly developing new strategies to evade detection. Additionally, the dataset’s features, while com- prehensive, may not encompass all aspects of phishing emails,suchasthevisualelements(e.g.,logos,formatting) often used to deceive users. Despite these limitations, the models developed in this project, particularly the LSTM, demonstrate strong potential for real-world application, provided they are continuously updated with new data and features.

# CONCLUSION

1. *Summary*

This project successfully demonstrated the effective- ness of a multi-modal approach to phishing email de- tection, combining text-based and URL-based features to achieve high classification accuracy. The LSTM model, when trained on preprocessed data, outperformed all other models,achievinganaccuracyof0.9828,precisionof

0.9896, recall of 0.9791, F1-score of 0.9941, and ROC- AUC of 0.9946. These results highlight the potential of deep learning techniques in addressing complex cyberse- curity challenges. Traditional models like XGBoost also performedwell,particularlyafterpreprocessing,underscor- ing the importance of data preparation in machine learning workflows. The visualizations, including the correlation heatmap, word clouds, and confusion matrices, provided valuable insights into the dataset and model performance, confirming the relevance of features like ‘numurls‘ and ‘bodylength‘ in detecting phishing emails.

The project also highlighted the impact of preprocessing on model performance, as evidenced by the consistent improvement in metrics across all algorithms when using preprocessed data. By addressing class imbalance through carefuldatasplittingandleveragingadiversesetofmodels, we ensured a robust evaluation of our approach. The findings of this study contribute to the broader field of cybersecuritybyofferingareliableframeworkforphishing email detection that can be further refined and deployed in real-world settings.

1. *FutureWork*

Futureresearchcanbuildonthisworkbyexplor-ing several promising directions. First, the adoption of advanced neural architectures, such as transformer-based models(e.g.,BERT)orattentionmechanisms,couldfurther enhance detection accuracy by capturing more nuanced patterns in email text. These models have shown remark- able success in natural language processing tasks andcould provide a significant boost to phishing detection capabilities.Second,incorporatingadditionalfeatures,such as pre-trained word embeddings (e.g., GloVe, Word2Vec) or contextual analysis of URLs (e.g., examining the rep- utation of domains), could improve model robustness and generalization to new types of phishing attacks.

Another avenue for future work is the integration of visual features, such as the analysis of email formatting, embedded images, or logos, which are often used in phishingemailstodeceiveusers.Techniqueslikecomputer vision could be employed to extract these features and combine them with text-based features in a multi-modal framework. Additionally, deploying the models in a real- world email server or security system would allow for practical evaluation and refinement, addressing challenges likelatency,scalability,andadaptabilitytonewthreats.To ensure long-term effectiveness, continuous monitoring and periodic retraining of the models with updated phishing data are essential, as attackers are likely to evolve their tactics over time. Finally, exploring techniques to handle classimbalancemoreeffectively,suchasoversamplingthe minority class (legitimate emails) or using cost-sensitive learning, could further improve model performance and fairness.