جامعة الملك سعود
King Saud University

كلية علوم الحاسب والمعلومات
قسم تقنية المعلومات

Course Project

PhishSpotter

Student ID	Lecture Section	Name
443201045	56702	Lina Alharbi
443200591	56702	Leen Aqahtani
443200581	56702	Jaida Alfadda
443203037	56702	Dana Alomar
443203869	56702	Aafia Ghulam Muhammad

2024 / IT 461

Supervised by:

Dr. Abeer Aldayel

Table of Contents

Table of Figures

Table of Tables

1. Introduction

Phishing websites are increasingly common and dangerous, as they deceive users into revealing sensitive information like passwords and financial details. These websites often mimic legitimate sites, making it difficult for users to recognize them as fraudulent. With the growth of online transactions, detecting phishing websites has become essential for online security.[1]

1.1 Problem Background

Phishing detection is a critical area within cybersecurity, as traditional methods like blacklists are often slow to adapt to new phishing tactics. Machine learning offers a proactive approach by identifying patterns in features of phishing websites, enabling detection of new threats without relying on predefined lists.

1.2 Project Goal

Our project aims to develop a machine learning model to detect phishing websites based on specific characteristics extracted from URLs and domain information. The model's input includes a vector of 30 features, such as having_IP_Address, having_At_Symbol, and pop-up windows and Iframe use. This process is illustrated in Figure 1

Evaluation Metrics

We will assess model performance using:

- **Accuracy**: Overall correctness.
- **Precision**: Accuracy in identifying phishing sites.
- **Recall**: Effectiveness in detecting all phishing sites.
- **F1 Score**: Balance between precision and recall, particularly useful for class imbalance.

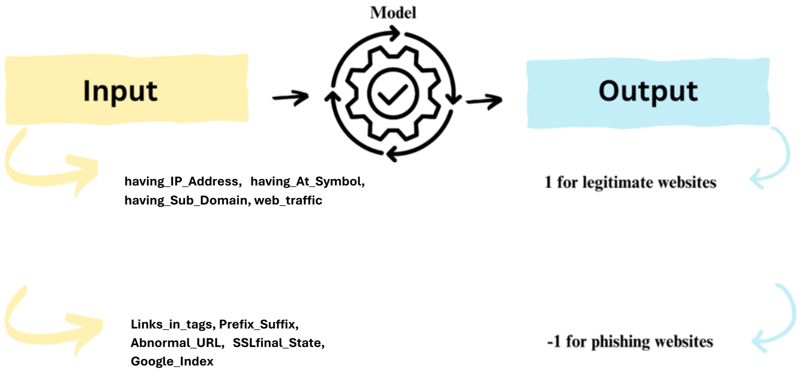

Model
Input
Output
having_IP_Address, having_At_Symbol,
having_Sub_Domain, web_traffic
1 for legitimate websites
Links_in_tags, Prefix_Suffix,
Abnormal_URL, SSLfinal_State,
Google_Index
-1 for phishing websites

Figure 1. Illustration of phishspotter model

2. Related Works

In the technologically advanced world, phishing websites are becoming increasingly common. They trick the users by imitating legitimate websites making it hard for those users to distinguish between fake and legitimate so that they reveal sensitive personal information such as passwords and financial details. Therefore, it is crucial to find a proper solution for phishing website detection to ensure user safety and protect data integrity. Traditional methods for checking such as blacklists are often not too reliable to keep up with the evolving phishing tactics. Machine Learning and Deep Learning offers dynamic solutions by analyzing those trends very quickly without having to rely on blacklists.

The objective of our project is to develop a machine learning model to detect phishing websites based on the characteristics of the URLs of the searched website. We will be using RNN, SVM and LR algorithms to classify whether the browsed website is legitimate or fake. Research was done on previous related work to compare and achieve better models which are written down below.

1. 2. In this paper [2], they have explored how well to classify phishing URLs from the given set of URLs containing benign and phishing URLs. They use feature extraction using lexical analysis which includes domain names subdomains and other components, they trained different classifiers e.g. Logistic Regression, Naïve Bayes Classifier, Random Forest, Decision Tree and KNearest Neighbor. The performance metrics show that all classifiers are suitable for the phishing URL detection tasks, the classifiers Random Forest and Gaussian Naïve Bayes classifiers result in better accuracies of about 98%. The Area Under the Curve (AUC) values were similar for all classifiers, but Gaussian Naïve Bayes had the highest AUC of 0.991, making it the most suitable model for phishing URL classification in this experiment.

2. This study is about phishing website detection [3], it compares 16 machine learning classifiers using semantic features such as URL and domain identity, HTML, and JavaScript characteristics. The best classifiers, Gradient Boosting and Random Forest, achieved about 97% accuracy. Meanwhile, models like GaussianNB and Stochastic Gradient Descent underachieved. The research demonstrates the effectiveness of semantic features in classifying phishing websites and highlights the growing threat of phishing attacks due to increasingly advanced techniques.

3. The paper's [4] author tried to address phishing attacks, an increasingly concerning problem in cybersecurity. The document focuses on identifying phishing attacks through machine learning (ML) methodologies. The author's proposed the development of a model using machine learning algorithms, Random Forest (RF) and Decision Tree (DT), for the detection of phishing websites. The dataset used covered many features related to phishing websites. Feature selection approaches, including Principal Component Analysis (PCA), have been used to enhance accuracy and minimize unnecessary data. To solve this problem, the author created a model using two machine learning algorithms, Random Forest and Decision Tree, to classify phishing websites. For the feature selection, PCA was performed to identify significant features from the dataset, hence minimizing redundancy. The model's accuracy and performance were assessed by a confusion matrix, which

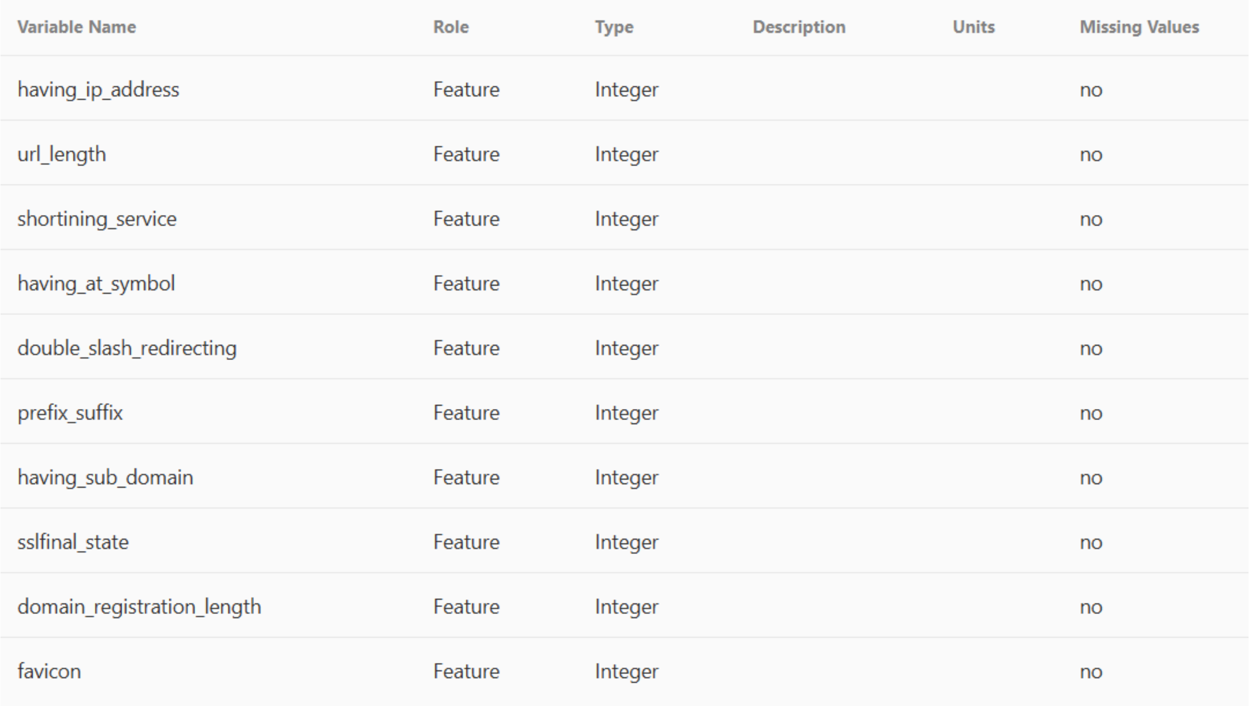
measured accuracy, precision, recall, and F1 score. Random Forest attained an accuracy of 97%, which shows that the model is more capable of managing overfitting compared to Decision Trees.

3. Data

Our project utilizes a dataset of 11,055 instances, each represented by 30 attributes that capture key characteristics of websites, enabling effective differentiation between legitimate and phishing sites. This dataset, sourced from Kaggle [5], was selected for its comprehensiveness and diversity, offering a rich collection of data points crucial for training a highly accurate phishing detection model.

The significance of this dataset lies in its role in combating the growing threat of phishing attacks, which continue to cause substantial financial and data losses worldwide. By leveraging this resource, our project aims to enhance web security by improving the detection and prevention of phishing attempts. This, in turn, helps protect individuals and organizations from cyber threats, safeguarding their assets and sensitive information.

The dataset serves as the backbone of our project, providing the foundation for developing and fine-tuning our machine learning model. Its detailed attributes are instrumental in ensuring the model's ability to accurately and reliably identify phishing websites. Table 1 presents the summary statistics of the dataset subset used in this study, highlighting its relevance and utility in achieving our project goals.



Variable Name	Role	Type	Description	Units	Missing Values
having_ip_address	Feature	Integer			no
url_length	Feature	Integer			no
shortining_service	Feature	Integer			no
having_at_symbol	Feature	Integer			no
double_slash_redirecting	Feature	Integer			no
prefix_suffix	Feature	Integer			no
having_sub_domain	Feature	Integer			no
sslfinal_state	Feature	Integer			no
domain_registration_length	Feature	Integer			no
favicon	Feature	Integer			no

Figure 2.dataset

This dataset is vital for our detection of phishing websites as it provides us with most of the features needed to achieve our goal. The dataset was preprocessed by checking if there were any missing values and outliers to deal with them. This step was taken to smoothen and fasten the process of machine learning. Table 1 shows the summary of the dataset details in a more organized way.

	Information
Number of data instances	11055
Number of features	30
Type of features available	Integer
Number of classes	2 (Phishing or legitimate)

Table 1.dataset summary

4. Methods

The primary goal of PhishSpotter is to leverage machine learning models to detect phishing websites, thereby improving online security. This solution combines Logistic Regression (LR), Support Vector Machine (SVM), Recurrent Neural Network (RNN), and Convolutional Neural Network (CNN) to analyze various indicators in website attributes and detect potentially harmful phishing sites.

4.1 Machine learning models

1. **Logistic Regression (LR):**
   Logistic Regression was used as a baseline due to its simplicity and interpretability. It sets a foundational benchmark for the project, allowing more complex models to be evaluated against a simple, linear approach. Despite its simplicity, LR often provides competitive results for binary classification tasks like phishing detection.

2. **Support Vector Machine (SVM):**
   The SVM model, particularly with an RBF kernel, was chosen for its ability to capture complex, non-linear relationships. SVM maximizes the margin between phishing and legitimate websites, which helps it generalize well. By leveraging kernel tricks, it can model high-dimensional patterns inherent in phishing websites, making it a powerful tool for this classification task.

4.2 Deep learning model

Recurrent Neural Network (RNN):
An RNN with LSTM layers was implemented due to its effectiveness in capturing sequential patterns. Certain features in URLs and domain information have sequential dependencies that the RNN can interpret, making it valuable in analyzing structured and time-based data patterns associated with phishing.

4.3 Feature Engineering and Preprocessing

1. **Data Preprocessing:**
   Features were standardized using StandardScaler to ensure uniform input to all models, which improves model convergence and performance. The dataset was split into training and test sets to enable model evaluation and avoid overfitting.

2. **Hyperparameter Tuning and Cross-Validation:**
   For Logistic Regression and SVM, GridSearchCV was used to optimize hyperparameters, particularly for the SVM model with an RBF kernel, which showed the highest performance in earlier evaluations. The RNN model leveraged K-Fold Cross-Validation to find the optimal combination of hyperparameters, enhancing its robustness against overfitting. CNN,

benefiting from its convolutional architecture, directly processed the standardized data without requiring complex sequential features, proving efficient for phishing detection.

3. **Evaluation Metrics:**
   Area Under the Curve (AUC) was used as a key evaluation metric for all models. Logistic Regression, SVM (RBF), RNN, and CNN were evaluated and compared based on AUC scores and other performance metrics (accuracy, precision, recall, and F1-score). Among the models, SVM with an RBF kernel achieved the highest AUC score, indicating a strong capability for phishing detection. CNN also showed promising results, justifying its inclusion as a valuable model for this project.

5. Experiment

To find the optimal algorithms to build our model. We began by addressing redundancy and missing values. A redundant column Index was removed, and no missing values were identified in the dataset. Outlier detection using z-scores revealed significant outliers in the RightClick and Iframe features. These outliers were retained, as they represent potential phishing behaviors. The dataset was split into training and testing sets using an 80:20 percentage. For deep learning models, 20% of the training set was further set aside for validation purposes. This setup ensured sufficient data for training, validation, and evaluation while maintaining balanced class distributions.

5.1 LR

We initially chose logistic regression (LR) as our initial model for phishing detection due to its simplicity, interpretability, and effectiveness in binary classification tasks. Before training the data, the dataset was normalized using StandardScaler to scale all features to a uniform scale. This preprocessing step ensured that no single feature dominates the learning process, which is particularly important for models like LR. Since the dataset primarily consists of numeric and categorical features, data augmentation techniques were not applicable for this task.

To optimize the logistic regression model, we used GridSearchCV, which systematically tested different values of the regularization parameter (C). The parameter controls the balance between fitting the training data well and maintaining generalization to unseen data. The grid of parameters included values for C ranging from [0.01, 0.1, 1, 10, 100], regularization types (penalty) of ['l1', 'l2'], and the 'liblinear' solver.

Methodology was used a 5-fold cross-validation approach was employed during the grid search to evaluate the model's performance on each set of parameters. This approach ensured that the selected hyperparameters were robust and did not overfit the training data.

The optimal parameters identified were C = 0.1, penalty = 'l1', and solver = 'liblinear' which struck the optimal balance between underfitting and overfitting.

Using these parameters, the model achieved an accuracy of 92.72%, a precision of 92.87%, a recall of 94.42%, and an F1 score of 93.64% on the test set. These metrics demonstrated that logistic regression could effectively distinguish between phishing and legitimate sites.

Further analysis was conducted through a classification report and a confusion matrix, revealing high performance across all classes. The classification report showed weighted averages for precision, recall, and F1-score at 93%, while the confusion matrix highlighted accurate prediction distribution with minimal misclassifications.

The experiment relied on SciKit-Learn to implement the logistic regression, StandardScaler, and GridSearchCV, while Matplotlib and Seaborn were used to visualize the confusion matrix and classification metrics. All training and evaluation were performed on CPU, highlighting the efficiency of logistic regression as a lightweight yet effective baseline model for phishing detection.

5.2 SVM

Our next model is SVM, first we scaled the data using *StandardScaler.* Since this classification task did not involve image or sequential data, data augmentation techniques were unnecessary. We experimented three different classifiers with three different kernel functions: RBF, Polynomial, and Linear. The RBF kernel demonstrated the best performance among the three, so we selected it for further tuning.

After that, *GridSearchCV* was applied to fine-tune the C and gamma parameters for the RBF kernel:

- **Parameter grid**: C values of [1, 10, 100] and gamma values of [0.01, 0.1, 1].
- **Methodology**: A 5-fold cross-validation method was used within the grid search to assess model performance.

The optimal parameters found were:

- **C = 100** and **gamma = 0.1**. These values optimized the RBF model's performance by balancing the margin and minimizing misclassifications.

After parameter tuning, we evaluated the model’s performance using accuracy and F1 score metrics. the best tuned RBF model achieved an accuracy of 96.65% and an F1 score of 96.65%. Additionally, we printed a classification report and a confusion matrix to gain insights into the model’s prediction distribution across classes.

The main libraries used during this experiment includes SciKit-Learn's modules for SVM, StandardScaler, along with Matplotlib and Seaborn for visualization. All training and tuning processes were executed on a CPU.

5.3 RNN

To explore the sequential aspects of the dataset, we utilized an **LSTM-based RNN**. The model architecture included:

- An **LSTM layer** with 64 units.
- **Dropout layers** with a rate of 0.3 to prevent overfitting.
- A **Dense layer** with sigmoid activation for binary classification.

We employed **K-fold cross-validation** to tune hyperparameters, including the number of LSTM units, dropout rates, optimizers (adam, rmsprop), epochs, and batch sizes. Despite these efforts, the RNN model achieved:

- **Accuracy**: 86.70%
- **F1 Score**: 87.38%.

While the RNN benefited from GPU acceleration for faster training, it was less effective compared to the SVM model for this dataset, highlighting the RBF kernel's superiority in handling non-sequential features.

5.5 Evaluation Metrics

To evaluate the models, we utilized metrics such as accuracy, precision, recall, F1 score, and AUC. SVM with the RBF kernel emerged as the best-performing model, Logistic Regression followed closely. While RNN showed reasonable results but was less effective compared to the other models for this specific task.

5.6 Regularization and Generalization Techniques

Regularization techniques like L1 and L2 penalties in Logistic Regression and hyperparameter tuning in SVM helped enhance model generalization. The RNN dropout layers, reducing overfitting and improving robustness.

6. Results and Discussion

1-Logistic regression

The confusion matrix for Logistic Regression indicates that the model performs well overall, correctly identifying 865 phishing websites and 1,185 legitimate websites. However, it misclassified 91 phishing websites as legitimate and 70 legitimate websites as phishing. While these results show that the model is effective at distinguishing between phishing and legitimate websites, the false negatives suggest room for improvement.

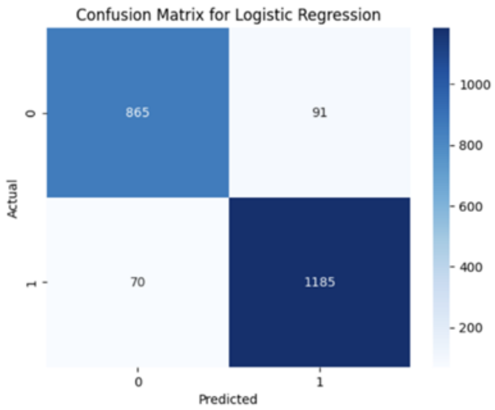


Figure 3 LR

The ROC curve for Logistic Regression shown in Figure 4 shows that the model does a good job of distinguishing between phishing and legitimate websites. The curve rises quickly towards the top-left corner, which means the model has a high true positive rate while keeping the false positive rate low. The AUC score of 0.9769 indicates that the model performs well overall. However, the curve flattens a bit at the top, showing that the model isn't perfect and might miss some phishing websites or misclassify some legitimate ones. While it's a strong model, there's still room to improve, especially compared to other models like SVM.

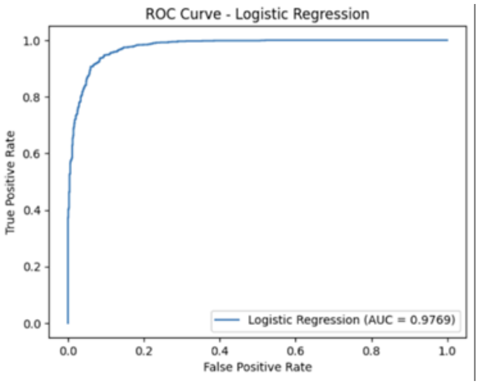


Figure 4.LR AUC

Logistic Regression performs reliably as a baseline model, achieving competitive precision and recall. However, its inability to handle non-linear relationships limits its performance compared to SVM.

2-SVM

The performance of three SVM kernels—RBF, polynomial, and linear—was evaluated to identify the most suitable kernel for phishing detection. Table 2 presents the metrics for each kernel on the test set.

Kernel	Accuracy
Polynomial	95.02%
RBF	96.65%
Linear	92.85%

Table 2.SVM kernels

The RBF kernel achieved the highest accuracy at 96.65%, demonstrating its superior ability to model complex, non-linear relationships in the phishing detection dataset. The polynomial kernel followed with an accuracy of 95.02%, performing competitively but falling short of the RBF kernel due to its tendency to overfit higher-degree relationships. The linear kernel performed the worst, with an accuracy of 92.85%, which is expected given its inability to handle the intricate, non-linear patterns inherent in phishing detection.

The confusion matrix for the best SVM model, which uses the RBF kernel, is shown in Figure 5**.** The model correctly identified 909 phishing websites and 1228 legitimate websites, demonstrating its high accuracy in detecting phishing. These results highlight the RBF kernel's ability to recognize phishing patterns effectively. There were only 47 phishing websites incorrectly classified as legitimate (false negatives) and 27 legitimate websites wrongly classified as phishing (false positives). The low false negative rate is especially important because it ensures that most phishing websites are detected, which helps protect users. While there are some false positives, this is less of a problem since flagging a legitimate site by mistake is not as serious as missing a phishing attack. Overall, the RBF kernel strikes a good balance between accurate detection and minimal errors, making it highly effective for phishing detection.

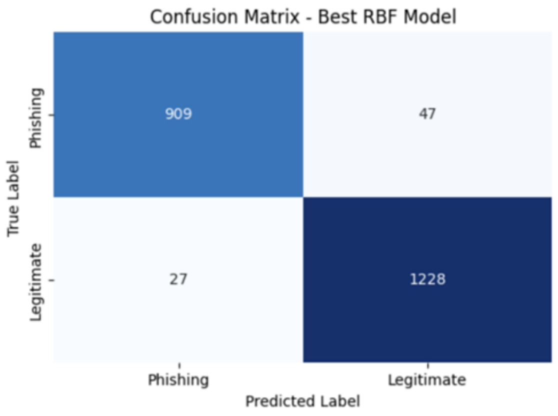


Figure 5.SVM

The SVM model with RBF kernel had the highest AUC score among the machine learning models, showcasing excellent discriminatory power. The curve indicates near-perfect classification performance, reflecting the model's ability to minimize false positives and false negatives.

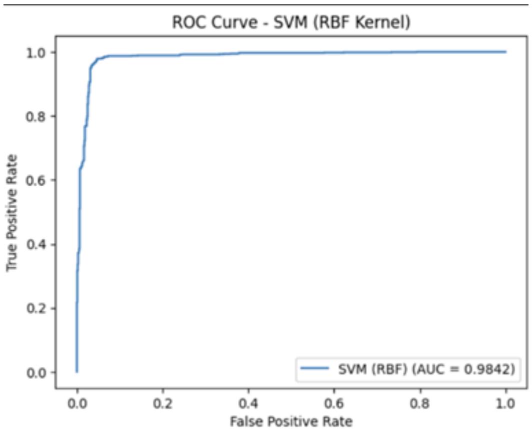


Figure 6.SVM AUC

3-RNN

The accuracy of the RNN model was 86.7%, which was lower compared to both Logistic Regression and SVM. The model's precision and recall also lagged behind, reflecting its difficulty in consistently classifying phishing and legitimate websites. The precision, which measures how often phishing predictions were correct, was impacted by the RNN's tendency to misclassify legitimate websites as phishing. Similarly, the recall, which measures how well the model detected actual phishing websites, showed that it failed to identify a significant portion of phishing attempts. While the RNN demonstrated some ability to classify the data, its overall performance was not as strong as SVM or Logistic Regression, indicating that it struggled with the static, feature-based nature of phishing detection tasks.

The RNN model achieved an AUC score of 0.9105, as shown in Figure 7. The curve indicates moderate classification performance, with a less steep rise compared to models like Logistic Regression and SVM. This reflects a weaker ability to trade off between sensitivity and specificity, meaning the RNN struggled to consistently distinguish phishing websites from legitimate ones. While the AUC score suggests the model has some capacity for classification, it is clear that the RNN is less effective for this task. This limitation likely stems from its design for sequential data, which does not align well with the static nature of phishing detection. Compared to models like SVM, the RNN underperformed, emphasizing its unsuitability for this specific application.

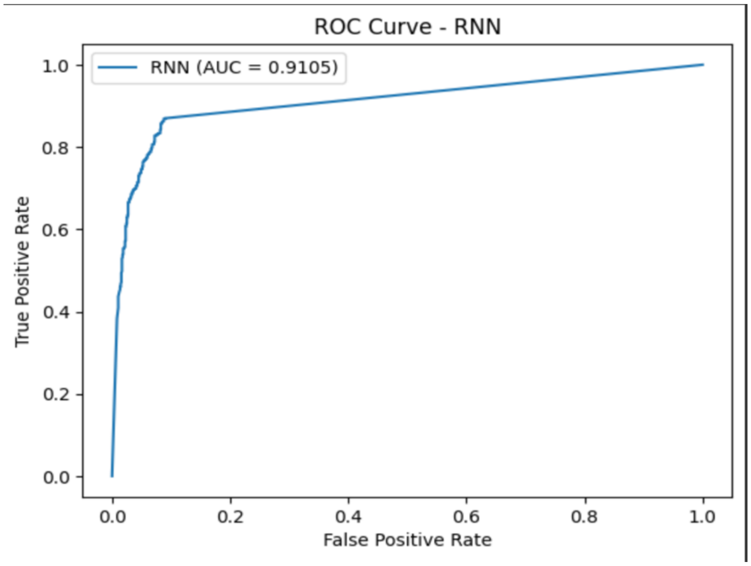


Figure 7.RNN AUC

Overall results

The results of the implemented models—Logistic Regression, SVM (RBF Kernel), RNN—were evaluated in terms of accuracy, precision, recall, F1 score, and AUC. These metrics summarize the models' ability to classify phishing websites accurately. A detailed comparison is presented in Table 3.

Model	Accuracy	Precision	Recall	F1 score	AUC
Logistic Regression	0.9272	0.9287	0.9442	0.9364	0.9769
SVM (RBF)	0.9665	0.9631	0.9785	0.9708	0.9842
RNN	0.8670	0.9470	0.8112	0.8738	0.9105

Table 3.OVERALL RESULTS

How well each model generalizes to unseen data:

Logistic Regression: Logistic Regression demonstrated good generalization to unseen data, achieving an accuracy of 92.72% and an AUC score of 0.9769, as seen in its ROC curve. The steep rise in the ROC curve indicates a high true positive rate with a low false positive rate, reflecting the model's ability to effectively distinguish between phishing and legitimate websites. However, as shown in its confusion matrix, the model misclassified some phishing websites as legitimate (false negatives) and a few legitimate websites as phishing (false positives). Despite this, its balanced precision (92.87%) and recall (94.42%) indicate that Logistic Regression is a reliable baseline model with reasonable performance on unseen data.

SVM (RBF Kernel): The SVM model with the RBF kernel exhibited the best generalization among all models, with an accuracy of 96.65% and an AUC score of 0.9842. The ROC curve shows an almost perfect rise toward the top-left corner, indicating its exceptional ability to separate phishing websites from legitimate ones on unseen data. The confusion matrix supports this, showing very few false negatives and false positives, with a recall of 97.85%, demonstrating its strength in detecting phishing attempts. Its superior performance in both precision (96.31%) and F1 score (97.08%) further highlights its robustness and reliability in generalization.

RNN: The RNN model struggled with generalization, achieving a lower accuracy of 86.70% and an AUC score of 0.9185. The ROC curve, while moderately steep, does not rise as sharply as those of Logistic Regression or SVM, reflecting the RNN's challenges in distinguishing phishing websites from legitimate ones. The recall (81.12%) was significantly lower, suggesting that the model missed many phishing websites, while the precision (94.70%) indicates it was conservative in flagging phishing attempts. These results suggest that the RNN's architecture, optimized for sequential data, was less effective for this static dataset, limiting its generalization capability.

Findings and Insights

- The SVM model with the RBF kernel stood out as the best-performing model, achieving the highest accuracy (96.65%) and AUC score (0.9842). Its ability to model non-linear relationships in the dataset enabled it to detect phishing websites with minimal false positives and false negatives. This highlights that complex, non-linear kernels like RBF are highly effective for tasks like phishing detection.

- Logistic Regression, despite being a simpler model, demonstrated strong performance with an accuracy of 92.72% and an AUC score of 0.9769. Its reliability as a baseline model is notable, especially for applications requiring less computational power. However, its inability to capture non-linear relationships limited its overall effectiveness compared to SVM.

- The RNN model struggled to generalize well, achieving lower metrics across the board (accuracy of 86.7%, AUC of 0.9185). This suggests that RNNs, designed for sequential data, are not ideal for phishing detection tasks where data relationships are static. This insight emphasizes the importance of matching model architecture to the data characteristics.

- While SVM (RBF) achieved a high recall of 97.85%, minimizing the risk of missing phishing websites, it maintained a balanced precision of 96.31%. On the other hand, RNN had a higher precision (94.70%) than recall (81.12%), indicating that it was conservative in flagging phishing attempts but missed several actual phishing sites. Logistic Regression maintained a good balance between precision and recall, showcasing its strength in achieving consistent performance.

- The findings suggest that the SVM with RBF kernel is particularly well-suited for phishing detection tasks. Its ability to accurately model complex non-linear patterns in the data while maintaining a high recall is critical, as failing to detect phishing websites could have severe consequences for users. These insights underline the importance of selecting the right model for the task, with the SVM with RBF kernel emerging as a reliable choice due to its robustness and ability to generalize effectively to unseen data.

7. Conclusion

Our project explored the detection of phishing websites using three machine learning models—Logistic Regression, Support Vector Machine (SVM) with RBF kernel, and Recurrent Neural Network (RNN)—evaluated across accuracy, precision, recall, F1 score, and Area Under the Curve (AUC).

Our study found that the SVM with RBF kernel was the most effective model for phishing detection, achieving an accuracy of 96.65% and an AUC score of 0.9842. Its ability to handle complex, non-linear relationships allowed for high recall and precision, ensuring minimal false positives and negatives. Logistic Regression provided reliable baseline performance with an accuracy of 92.72% and an AUC score of 0.9769, demonstrating its practicality for computationally constrained applications. Conversely, the RNN underperformed with an accuracy of 86.7% and struggled with recall, highlighting its limitations in static, feature-based datasets.

One of the primary challenges we faced was the time-consuming nature of running the RNN model on the entire dataset, which was compounded by its suboptimal performance for this task. Additionally, Logistic Regression's inability to capture non-linear relationships limited its effectiveness compared to SVM.

To address these challenges and enhance future performance, we propose several improvements:

- **Ensemble Methods**: Combining the strengths of SVM and Logistic Regression through ensemble techniques could yield more robust results.
- **Real-Time Detection**: Implementing these models in real-time detection systems would allow dynamic adaptation to emerging phishing threats.
- **Dataset Augmentation**: Expanding the dataset with more diverse and complex phishing samples could enhance the models' ability to generalize to unseen data.
- **further optimization**: Techniques like Bayesian Optimization or Genetic Algorithms could be explored to fine-tune the parameters more systematically, potentially yielding even better performance.

In conclusion, this project demonstrated that the SVM with RBF kernel is highly effective for phishing detection, providing strong performance and generalization. While the models performed well overall, addressing the identified limitations and pursuing the proposed improvements would further advance the system's capabilities, ensuring better protection against evolving cyber threats.

8. Contributions

Member	**Role**	**Responsibilities**
Lina Alharbi	*Leader*	Introduction, method, Experiment
Jaida Alfadda	*Member*	Experiment, Conclusion
Dana Alomar	*Member*	Results and Discussion
Leen Alqahtani	*Member*	Experiment
Aafia Muhammad	*Member*	*Related works and Data*

Table 4.CONTRIBUTIONS

9. References

[1] ChatGPT, "Phishing websites are increasingly common and dangerous," OpenAI, Sep. 2024. [Online]. Available: https://chat.openai.com/

[2] J. Kumar, B. Rajendran, A. Santhanavijayan, B. Bindhumadhava, and B. Janet, “Phishing Website Classification and Detection Using Machine Learning,” 2020 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, Jan. 2020, pp. 1-6, doi: 10.1109/ICCCI48352.2020.9104162. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9375997.

[3] A. Almomani, M. Alauthman, M. T. Shatnawi, M. Alweshah, A. Alrosan, W. Alomoush, and B. B. Gupta, "Phishing website detection with semantic features based on machine learning classifiers: A comparative study," International Journal on Semantic Web and Information Systems (IJSWIS), vol. 18, no. 1, pp. 1-24, 2022. Available: https://www.igi global.com/article/phishing-website-detection-with-semantic-features-based-on-machine-learning-classifiers/297032

[4] M. N. Alam, D. Sarma, F. F. Lima, I. Saha, R. -E. -. Ulfath and S. Hossain, "Phishing Attacks Detection using Machine Learning Approach," 2020 Third International Conference on Smart Systems

and Inventive Technology (ICSSIT), Tirunelveli, India, 2020, pp. 1173-1179. [Online] .Available: https://ieeexplore.ieee.org/abstract/document/9214225.

[5] A. Kumar, "Phishing Website Dataset," Kaggle, 2020. [Online]. Available: https://www.kaggle.com/datasets/akashkr/phishing-website-dataset/data . [Accessed: Sep. 15, 2024].