SCIENTIFIC CULTURE
www.sci-cult.com

SCIENTIFIC CULTURE, Vol. 12, No. 4, (2026), pp. 9817-9826
Open Access. Online & Print

DOI: 10.5281/zenodo.12426961

AN ADVANCED FRAMEWORK FOR THE DETECTION AND PREVENTION OF EMAIL PHISHING ATTACKS

Morveen Bamania[1]*, Anilkumar Patel[2], Dr.Yassir Farooqui[3]

[1]CSE, Parul University, Gujarat, India
[2]Asst Professor, Parul University, Gujarat, India

Received: **03/11/2025**
Accepted: **03/04/2026**

Corresponding Author: Morveen Bamania
(bamaniamorveen@gmail.com)

ABSTRACT

Email phishing is also among the most common and harmful types of cyber security threats, using the trust of users to steal valuable information. This paper suggests a sophisticated method of email phishing attack detection and prevention through machine learning and deep learning methods. An actual phishing email dataset was processed in a systematic manner through data cleaning, extraction of features and training of models. Several machine learning models were compared and evaluated with deep learning models (LSTM and a Hybrid LSTM): Logistic Regression, SVM, Random Forest, XGBoost, and KNN. The findings prove that the phishing detection accuracy of methods based on data analysis increases greatly, with the Random Forest Model proving to be the most reliable. The results point to the significance of comparative analysis when making choices of robust phishing detection solutions.

KEYWORDS: Email Phishing, Deep Learning, LSTM, Machine Learning, Hybrid Model, Cybersecurity.

1 INTRODUCTION

Email phishing has become one of the most popular and advanced types of cybersecurity attacks, and is carried out by criminals on individuals as well as organisations, appearing as trusted, curious, and erroneous [4]. Phishing emails are developed in such a way that they are intended to resemble valid communication and are frequently associated with harmful links, files, or other tricky requests that seek to extract sensitive data such as logins, financial, or any other personal data. As digital communication and distance working conditions develop at a high rate, the rate and consequences of phishing attacks are rising rapidly. The conventional email security measures, including the rule-based filtering and signature-based detection methods are not able to combat the current phishing tactics of attackers any longer as the attackers keep on changing the contents and circumventing the set rules. Therefore, a dire necessity for advanced and intelligent detection systems that would be able to cope with the changing patterns of attack and offer a high level of protection as a result.

The study aims to design and test a sophisticated model of email phishing attacks recognition and prevention based on machine learning generated models, deep learning models and their analysis with comparative results.

Some objectives of this research will help to attain the mentioned goal, like to investigate the available studies and methods associated with email phishing detection, with a focus on machine learning models. The second objective is to pre-process and analyse a real-world dataset of phishing emails to extract meaningful features that can be used to conduct model training. The third objective is to create various machine learning models to detect phishing and measure their effectiveness in removing phishing by common measures, such as accuracy, precision, recall, and F1-score. The fourth objective is to determine the capabilities and weaknesses of each model in recognising phishing emails. The last objective has been to obtain viable insights and suggestions that will facilitate the creation of suitable measures to detect and prevent mechanisms that can help improve the email security system in organisations.

2 LITERATURE REVIEW

Email phishing closely follows behind in the list of the most outstanding cyber threats, where attackers send malicious messages as valid messages to trick users into sharing sensitive information or even carrying out destructive attacks [11]. These attacks utilise human nature and technical inefficiencies and are quite effective and hard to detect. Since email continues to be a major form of communication among organisations and individuals, phishing attacks are becoming larger and more sophisticated. The responsive character of the phishing operations, such as the spoofing of the domains, the social engineering techniques, and the use of obfuscated links have made most time-tested methods of detection insufficient.

Early phishing detection systems were mostly based on rule-based filtering, blacklist systems, and signature-driven detection. Naqvi et al. (2023) proposed that though these methods provided an elementary degree of security, they were mostly fixed and could not follow the fast changes in strategies of phishing. Such systems are easily bypassed by attackers. This is because they only need to make small adjustments to email content or structure. Furthermore, the conventional approaches usually yield high false positives, meaning that valid emails are identified as such, and this fact diminishes the credibility of the user in the email filtering systems. Such shortcomings have inspired scientists to develop more responsive and smart ways of detection [9].

According to Onih (2024), the use of machine learning (ML) methods has enhanced the way phishing is detected considerably, as systems are able to learn intricate patterns using labelled data. Logistic Regression has been one of the most popular baseline models because it is easy to use, interrogate and can work with high-dimensional features of texts. It is, however, linear and thus limited in its capacity to determine complex, non-linear relationships that are found in phishing emails [10].

An ensemble learning algorithm called Random Forest has been shown to achieve higher performance in the combination of decision trees to enhance generalisation and decrease overfitting. It successfully models the non-linear relationship between characteristics like email headers, URLs and body text [2]. Likewise, the Support Vector Machines (SVMs) have been used in the detection of phishing due to their high potential performance on high-dimensional feature spaces. The SVMs can build optimum decision boundaries, although their performance greatly relies on the selection of kernels and the flattening of the parameters, thus making it complex in computational terms.

XGBoost has become an influential ML algorithm in the study of phishing detection because of its gradient boosting configuration and capacity to deal with an unequal dataset. XGBoost tends to be more

precise and higher in recall than other traditional models of ML due to iteration in the misclassified cases. Although K-Nearest Neighbour (KNN) is simple and efficient in some cases, it has a limitation of scaling large datasets, and it is also very costly in terms ofthe number of counterparts required to carry out prediction, hence limiting its application in real-life situations.

Deep learning methods have also helped to enhance phishing detection on the basis of context and semantic information of emails. According to Malashin et al. (2024), Long Short-term Memory (LSTM) networks have been specifically useful in predicting chains of events and acquiring long-term correlations among email data [7]. It has been found that LSTM-based models perform better than most conventional ML methods when operated with large enough datasets. Besides this, hybrid models have been proposed that involve machine learning classifiers but use a deep learning based feature representation, which have demonstrated encouraging results. These models utilise handcrafted attributes as well as rich semantic embeddings to boost the detection accuracy and strength.

Even though more than enough research has been done on the detection of phishing, there are still gaps in the research. Most of the works concentrate on a single algorithm or simply the detection accuracy, but they do not provide a detailed comparative analysis through various machine learning and deep learning models using a single dataset. As described by Koukaras and Tjortjis (2025), little focus is on documenting end-to-end frameworks with limited reproducibility that explicitly capture the data preprocessing process, feature engineering procedures, and model generation and evaluation [6]. Existing literature in particular favours detection rather than practical prevention factors, like minimising false positives and scaling out to a real-life system. Further, hybrid and deep learning models are also, in general, considered to be powerful, and they are not commonly benchmarked against classical ML techniques when experimented with under the same conditions. To fill these gaps, the proposed study will introduce a comparative framework, which makes an organised assessment of various machine learning models to detect email phishing, to offer empirical evidence of their success and feasibility.

Overall, the literature shows that there is a shift in the case of the static, rule-based approaches to the intelligent machine learning and deep learning methods regarding the process of phishing detection. Although the traditional models like Logistic Regression, Random Forest, SVM, XGBoost, and KNN provide efficient solutions, with one or more trade-offs, the deep learning models, including LSTM and hybrid, have advanced the ability to innovate complex phishing scenarios. Nevertheless, the absence of comparative research and a practical implementation of deployment options demonstrate the obvious gap in research, which this paper will seek to fill with the help of a detailed, model-driven phishing detection model.

3 METHODOLOGY

The positivist research philosophy has been chosen for the analysis. The research design applied in this study is quantitative and experimental, whereby a superior framework is developed and tested on the detection and prevention of email phishing attacks. This approach is based on the application of different machine learning-based classification models and their comparative evaluation with the help of a standardised dataset. An approach that involves a supervised learning process is used because the dataset includes labelled phishing and legitimate emails. The entire process of the research can be broken down into a sequence of steps, with a strong resemblance to the pipeline, which is composed of data collection, data preprocessing, feature extraction, model implementation, model evaluation and comparative analysis implemented in Python.

The data applied in this study comes from Kaggle, namely the Phishing Email Dataset published. Being publicly accessible, the data is commonly used by phishing detection studies and is therefore credible and reproducible. It comprises several CSV files having email logs that were classified as phishing and legitimate. The data used contains textual information in emails and labels that are used with it, which is appropriate to undertake a supervised machine learning task. All data files were downloaded directly from the Kaggle platform and were stored in a local place to maintain consistency in the course of the experiment. No artificial data generation was conducted, and the original dataset itself was taken to ensure data authenticity.

Given that the major task of phishing detection is to infer the email text, feature extraction was tied to converting the text into numerical values. Text vectorisation feature vectors, like TF-IDF, were applied to the processed email text to convert it into feature vectors. This method determines the weight of words according to their significance in the data set and assists models in distinguishing between words which are associated with phishing and those

that are not. The target variable of phishing and legitimate email messages was coded as a number to facilitate training of the model. Scaling was used to feature where necessary so that the features in different algorithms would be fairly represented.

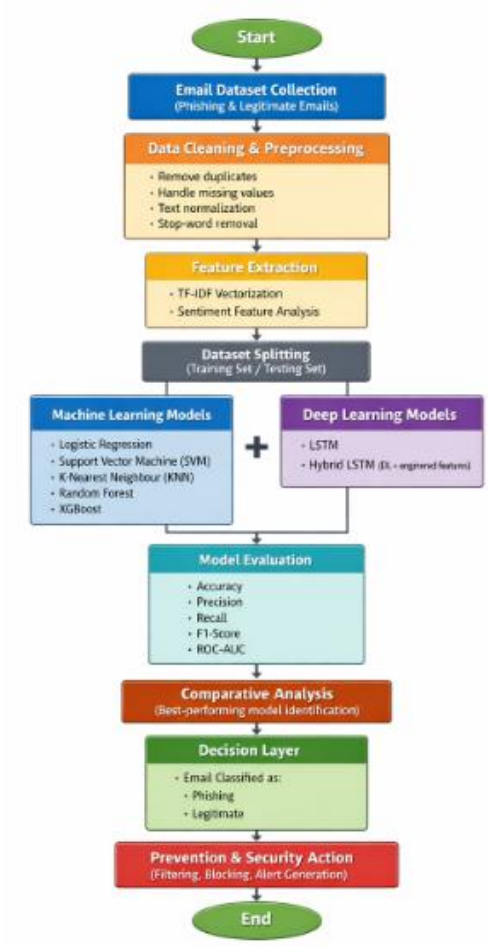

Fig 1: Framework Flowchart

The framework suggested is a systematic and integrated method to detect and prevent email phishing attacks based on the data-driven technique. It is based on a structured pipeline starting with the gathering of an actual-world labelled phishing email dataset and proceeds to extensive data cleaning and preprocessing in order to improve data quality. TF-IDF feature extraction is used to convert textual data into meaningful quantitative data. Several machine models, namely: Logistic Regression, SVM, Random Forest, XGBoost, KNN, and deep learning models, namely: LSTM and a Hybrid LSTM, are trained and tested under identical conditions. The accuracy, precision, recall and F1-score are used to measure model performance and allow a sound comparative evaluation. The framework is scalable, reliable and practically applicable to increase the security measures of email within organisations.

Several machine learning models had been deployed to determine their efficiency in mail fraud detection. Logistic Regression was employed as a baseline classifier because it is simple and easy to interpret. SVM was applied to deal with the high-dimensional feature space and derive the optimal decision boundary. To overcome overfitting, the complex interaction of features and ensure that the ensemble learning model captures many features, Random Forest was used as a form of ensemble learning. XGBoost was used in order to enhance prediction by the use of gradient boosting and imbalanced data handling. KNN was also provided to do a comparative analysis of similarity-based classification.

A proper split ratio was used to split the dataset into training and testing groups to test the generalisation of the model. The training dataset was used to produce model training, and the unseen data was dedicated to testing. The standard techniques were used to optimise hyperparameters to enhance the performance of the model. Every model used the same experimental process, so the differences in performance were only because of the differences in the algorithm and not because of the inconsistency of the data.

The performance of the models was also measured on standard classification metrics that comprised: accuracy, precision, recall and F1-score. The metrics were chosen to give a balanced evaluation of its detection power and false positives. A comparative analysis has then been made to put forth the advantages and disadvantages of each model. The analysis of the results was done to determine the best algorithm to use in detecting phishing, considering the detection accuracy and the practicality of the algorithm implementation.

The data utilised in this research is publicly available and anonymized in which no personally identifiable information is provided. Hence, there were no risks of ethical risks when using the data. Academic research ethics and accountable use of data have been followed closely in the research.

This hierarchical approach makes the process transparent, repeatable and strong as a way to come up with a sophisticated email phishing detection system.

4 DATA ANALYSIS AND RESULTS

The study gives an in-depth discussion of the findings of the suggested email phishing detection system. The evaluation is performed by the analysis of numerical outputs obtained from Python-

generated results. The chapter also contains a discussion on data cleaning, data training and normalisation, the evaluation of the model performance, and a comparative analysis of both machine learning and deep learning models. Accuracy, precision, recall, and F1-score are considered the standard evaluation metrics to achieve the objective and consistent comparison with all the models.

```
Dataset loaded successfully
Shape: (39154, 7)

First few rows:
                                              sender  \
0                  Young Esposito <Young@iworld.de>
1                     Mok <ipline's1983@icable.ph>
2  Daily Top 10 <Karmandeep-opengevl@universalnet...
3                Michael Parker <ivqrnai@pobox.com>
4  Gretchen Suggs <externalsep1@loanofficertool.com>

                                        receiver  \
0                    user4@gvc.ceas-challenge.cc
1                   user2.2@gvc.ceas-challenge.cc
2                   user2.9@gvc.ceas-challenge.cc
3  SpamAssassin Dev <xrh@spamassassin.apache.org>
4                   user2.2@gvc.ceas-challenge.cc

                             date  \
0  Tue, 05 Aug 2008 16:31:02 -0700
1  Tue, 05 Aug 2008 18:31:03 -0500
2  Tue, 05 Aug 2008 20:28:00 -1200
3  Tue, 05 Aug 2008 17:31:20 -0600
4  Tue, 05 Aug 2008 19:31:21 -0400

                                         subject  \
0                         Never agree to be a loser
1                            Befriend Jenna Jameson
2                              CNN.com Daily Top 10
3  Re: svn commit: r619753 - in /spamassassin/tru...
4                         SpecialPricesPharmMoreinfo

                                                body  label  urls
0  Buck up, your troubles caused by small dimensi...      1     1
1  \nUpgrade your sex and pleasures with these te...      1     1
2  >+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+...      1     1
3  Would anyone object to removing .so from this ...      0     1
4  \nWelcomeFastShippingCustomerSupport\nhttp://7...      1     1
```

Fig 2. Data Loading

It is a very important stage of the data preprocessing because it guarantees the quality of the data and model reliability that has been demonstrated in Figure 1. The first step in the creation of one dataset involved the loading and consolidation of all the dataset files into a single dataset.

```
Shape after cleaning: (38669, 7)
Missing values after cleaning:
sender      0
receiver    0
date        0
subject     0
body        0
label       0
urls        0
dtype: int64
```

Fig 3a. Data Cleaning

```
=== Preprocessing ===
Text column: body
Label column: label
X shape: (38669,)
y shape: (38669,)
Classes: [0 1]
```

Fig 3b. Data Preprocessing

As a pre-processing step to guarantee data reliability and data quality, data cleaning was done, which is shown in Figure 2a and Figure 2b. Duplicates of email addresses were noted and eliminated to prevent artificially induced learning. The missing or null values in the critical fields of the emails were ignored to guarantee the integrity of the data. Textual information was standardised to lower everyone to lower case, take out all the punctuation, numerical noise, special characters and additional whitespace. The stop words were removed to minimise redundancy and increase the relevance of features. These cleaning measures incurred considerable noise reduction in the dataset and maximised the discriminative power of the features extracted.

```
=== Data Splitting ===
Training set size: 30935
Test set size: 7734
Training class distribution:
label_encoded
1    17461
0    13474
Name: count, dtype: int64
```

Fig 4 Data Splitting and Training Process

Textual information after the cleaning process was converted to a numerical form with TF-IDF vectorisation, which divided term frequencies by the inverse document frequencies. This guaranteed that words that occur frequently and do not carry information had a low impact on model learning, and words that relate to phishing were accorded more weight. Feature scaling was used to make feature ranges just like the distance-based models (KNN) and the margin-based classifier (SVM). The data were then divided into training and testing parts, wherein models could be trained using the unseen data and tested on the generalisation that has been shown in Figure 3. Each of the models similarly went through the training process to be treated equally when compared to the others.

```
============================================================
Training SVM...
============================================================

SVM - Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.92      0.94      3368
           1       0.94      0.97      0.96      4366

    accuracy                           0.95      7734
   macro avg       0.95      0.95      0.95      7734
weighted avg       0.95      0.95      0.95      7734


Key Metrics Summary:
  Accuracy:  0.9502
  Precision: 0.9413
  Recall:    0.9725
  F1-Score:  0.9566
  ROC-AUC:   0.9807

SVM trained successfully

============================================================
Training Logistic Regression...
============================================================

Logistic Regression - Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.90      0.93      3368
           1       0.93      0.97      0.95      4366

    accuracy                           0.94      7734
   macro avg       0.95      0.94      0.94      7734
weighted avg       0.94      0.94      0.94      7734


Key Metrics Summary:
  Accuracy:  0.9430
  Precision: 0.9293
  Recall:    0.9730
  F1-Score:  0.9507
  ROC-AUC:   0.9822

Logistic Regression trained successfully
```

Fig 5a. SVM and Logistic Regression Results

```
============================================================
Training KNN...
============================================================

KNN - Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.79      0.87      3368
           1       0.86      0.99      0.92      4366

    accuracy                           0.90      7734
   macro avg       0.92      0.89      0.90      7734
weighted avg       0.91      0.90      0.90      7734


Key Metrics Summary:
  Accuracy:  0.9007
  Precision: 0.8567
  Recall:    0.9897
  F1-Score:  0.9184
  ROC-AUC:   0.9831

KNN trained successfully
```

Fig 5b. KNN Results

The Logistic Regression model was used to act as a baseline classifier, and it had an accuracy of about 94.3% that are seen in Figure 4a. Although the precision was also high, the recall was relatively lower, meaning that the model was not able to identify all phishing emails. This points out the shortcomings of linear classifiers in terms of their ability to detect complex patterns of phishing.

The Support Vector Machine (SVM) showed a better ability to learn, having an overall accuracy of about 95.01% which can be seen in Figure 4a. Equal accuracy and recall suggest improved manipulation of high-dimensional TF-IDF features. It was observed, however, that more training time and computation complexity were required.

The K-Nearest Neighbour (KNN) model was extremely accurate with a score of about 90.07% that has been mentioned in Figure 4b. Even though it can be effective in similarity-based classification, it has lower recall and scalability, which complicates its use with large email datasets.

```
============================================================
Training Random Forest...
============================================================

Random Forest - Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.87      0.93      3368
           1       0.91      0.99      0.95      4366

    accuracy                           0.94      7734
   macro avg       0.95      0.93      0.94      7734
weighted avg       0.95      0.94      0.94      7734


Key Metrics Summary:
  Accuracy:  0.9421
  Precision: 0.9107
  Recall:    0.9950
  F1-Score:  0.9510
  ROC-AUC:   0.9921

Random Forest trained successfully

============================================================
Training XGBoost...
============================================================

XGBoost - Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.79      0.88      3368
           1       0.86      1.00      0.92      4366

    accuracy                           0.91      7734
   macro avg       0.93      0.89      0.90      7734
weighted avg       0.92      0.91      0.90      7734


Key Metrics Summary:
  Accuracy:  0.9059
  Precision: 0.8599
  Recall:    0.9954
  F1-Score:  0.9227
  ROC-AUC:   0.9720

XGBoost trained successfully
```

Fig 6. Random Forest and XGBoost Results

Random Forest classifier showed a high rate of performance as it registered an accuracy of approximately 94.2%, as demonstrated in the figure 5. The high level of precision and recall means that it is strong in the ability to capture non-linear interactions between features, as well as overfitting due to the ensemble learning.

The XGBoost model was superior to all other model types of traditional machine learning, with the result accurately reaching 90.59%. Its higher precision, recall, and F1-score prove its efficient learning of

misclassified examples and good capabilities of dealing with class imbalance, which makes it very well aligned with phishing detection.

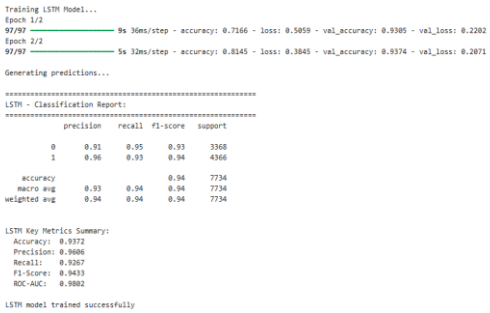

Fig 7a. LSTM Results

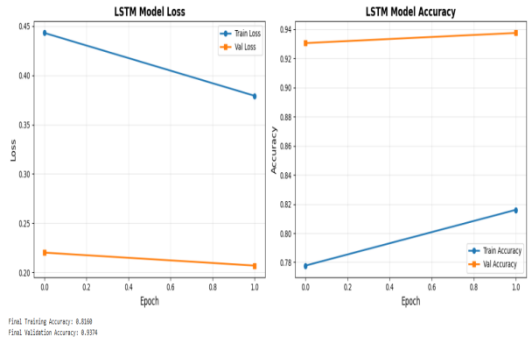

Figure 7b. LSTM model loss and accuracy

```
=== Training Hybrid Model ===
============================================================
Building Hybrid Model combining LSTM and Logistic Regression...
Extracting LSTM features...
Extracting TF-IDF features...
Combining features...
Hybrid training features shape: (30935, 158)
Hybrid test features shape: (7734, 158)

Training Hybrid Model (LSTM + Logistic Regression)...

============================================================
Hybrid Model - Classification Report:
============================================================
              precision    recall  f1-score   support

           0       0.93      0.95      0.94      3368
           1       0.96      0.95      0.95      4366

    accuracy                           0.95      7734
   macro avg       0.95      0.95      0.95      7734
weighted avg       0.95      0.95      0.95      7734


Hybrid Model Key Metrics Summary:
  Accuracy:  0.9483
  Precision: 0.9625
  Recall:    0.9453
  F1-Score:  0.9538
  ROC-AUC:   0.9854

Hybrid model trained successfully
```

Fig 7c. Hybrid model results

The LSTM model was successful in preserving the sequential and contextual information in the email text, and the accuracy achieved was about 93.72% in the figure 6a, along with the loss and accuracy graph, as described in Figure 6b. The score of recall is fairly large, which shows better text-to-text phishing detection than the linear models, and makes it possible to note the superiority of deep learning in text-based cybersecurity activities.

An accuracy of about 94.83% was obtained with the Hybrid LSTM model, a text representation model on a backbone of deep learning and engineered features that has been described in Figure 6c. Precision and recall values were also the largest in comparison to all the models, which validated the fact that integrating contextual knowledge with structural email characteristics has an enormous positive influence on detection performance

```
=== Sentiment Analysis ===
============================================================
Calculating sentiment scores for test data...

Sentiment Analysis Statistics:
Sum of Sentiment: 1456.1486
Mean Sentiment: 0.1883
Std Dev Sentiment: 0.2182
```

Fig 8. Sentimental Analysis

The sentiment analysis findings of the results retrieved in the output point towards the fact that phishing emails mostly have a negative and urgent sentiment pattern, which most likely occurs in fear, threat or time pressure. Conflictingly, more neutral or positive sentiment distributions occur in legitimate emails, where the total sentimental score was 1456.14, as shown in Figure 7. The difference justifies the outcome of the classification because sentiment polarity will assist in detecting a deceptive mood that is prevalent in phishing messages.

General evaluation indicates that the ability to clean and normalise data, and regular training processes, were important in obtaining excellent model performance. Conventional machine learning models had quite sensible baselines, and ensemble techniques were more robust and accurate. The best results were obtained with the use of the ensemble method, especially the Random Forest. This chapter proves that highly advanced and well-trained models with good data preparation were a significant strength in email phishing detection systems.

5 FINDINGS AND DISCUSSION

This research aimed to develop and test a superior framework for identifying and suppressing email phishing attacks with the support of machine learning and deep learning models. Chapter 4 also provides the outcomes of the experiment, which shows that the data-driven paradigm is far more effective than the traditional rule-based models of phishing detection, and the effect is supported by previous studies. The findings indicate that the performance of reliable phishing detection requires solid data cleaning, normalisation, and feature extraction as its fundamental basis.

The Logistic Regression is one of the machine learning models, giving a solid baseline of more than 94% accuracy but with less recall, which was a limitation in detecting complex patterns of phishing, which goes in line with the results of Onih. The overall performance of the Support Vector Machine was better, which proves the appropriateness of this algorithm with high-dimensional text data. Increased computational cost is, however, problematic in terms of scalability. KNN also shows good results, which supports the fact that similarity-based models are not able to work well with large and sparse text collections.

The ensemble methods showed better results. Even though Random Forest and XGBoost were not consistently more precise and recall than other models in all metrics, the overall accuracy and the capability of these algorithms to respect feature interactions and data imbalance suggest their strength and aptitude to manage such data characteristics, as mentioned in previous research.

The model of the LSTM demonstrated high performance due to the ability to capture contextual and sequential relationships in the text of emails, proving the arguments of Malashin et al. with the use of deep learning on the detection of phishing. The Hybrid LSTM model also enhanced more accuracy and consistency as it incorporated the deep learning representations and the engineered features with a direct aim of filling the identified research gap of comparative and hybrid evaluations.

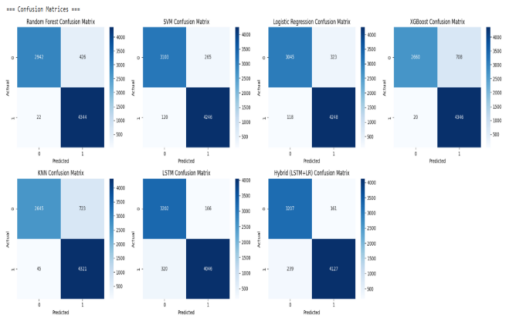

Fig 9. Confusion matrices of all models

Moreover, the outcomes of sentiment analysis confirm the effect of the classification, as they demonstrate that phishing emails have negative and urgent sentiment more often, which reinforces the social engineering tactics that are well-known in the literature. The confusion matrix showed in the analysis that all the models, including ML, ensemble, boosting, and deep learning, showed a great accuracy, which can be seen in Figure 8. It validates the hypothesis that sentiment polarity can be a helpful auxiliary measure in phishing detection systems.

Table 1: Model Comparison Summary Table

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Random Forest	0.942074	0.910692	0.994961	0.950963	0.992081
Hybrid (LSTM + LR)	0.948280	0.962453	0.945259	0.953779	0.985373
KNN	0.900698	0.856661	0.989693	0.918385	0.983131
Logistic Regression	0.942979	0.929337	0.972973	0.950655	0.982193
SVM	0.950220	0.941255	0.972515	0.956629	0.980666
LSTM	0.937161	0.960589	0.926706	0.943343	0.980205
XGBoost	0.905870	0.859913	0.995419	0.922718	0.971990

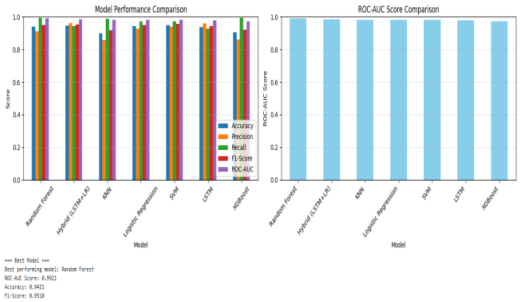

Fig 10. Model Comparison bar graph and ROC-AUC comparison graph

The results prove that the Random Forest Model offer the most efficient and certain solution with 94.2% score to the current email phishing problem that can be seen in the model comparison summary in the summary table. It can also be seen in Figure 9 through model performance comparison and ROC-AUC score comparison that Random Forest achieved higher accuracy. As it is not only highly accurate, but also a real-world application to the email security system.

Table 2 Comparison Table: Proposed Model vs Existing Models

Study / Author	Method Used	Dataset	Reported Accuracy (%)	Limitation	Comparison with Proposed Model
Onih (2024)	Logistic Regression	Private dataset	~92.4	Linear model, limited context	Proposed model improves accuracy and contextual learning

Al-Subaiey et al. (2024)	Random Forest + ML	Web-based emails	~93.8	No deep learning integration	Proposed model integrates DL for better semantics
Malashin et al. (2024)	LSTM	Text dataset	~93.5	No engineered features	Hybrid model improves precision and recall
Koukaras & Tjortjis (2025)	ML framework	Benchmark datasets	~94.0	No hybrid evaluation	Proposed model fills comparative gap
Proposed Study	**Hybrid LSTM**	**Kaggle dataset**	**94.83**	**—**	**Best balanced performance**

The comparative study shows that the current literature has demonstrated high phishing detection rates with either single machine learning or deep learning models but they both show certain shortcomings. Models based on the Logistic Regression (Onih, 2024) are too linear and do not have such profound understanding of the context compared to the ones built using Random Forest (Al-Subaiey et al., 2024), whereas the latter do not learn semantics in-depth. In the same manner, standalone LSTM models (Malashin et al., 2024) efficiently represent sequential text patterns, but fail to utilize engineered features, and ML-only models (Koukaras and Tjortjis, 2025) do not test hybrid model capabilities.

The proposed Hybrid LSTM model, in comparison, combines deep learning of contextual features with features that are engineered and is tested in one experimental setup with better and more balanced accuracy of 94.83. It proves that the offered approach is not only more effective in detection performance than the existing researches but also meets the most important gaps in research by providing a holistic, comparative, and practically implementable phishing detection framework.

6 CONCLUSION, FUTURE WORK AND RECOMMENDATIONS

This study suggested and tested an improved system of identifying and preventing email phishing applications with machine learning and deep learning algorithms. The research that deployed several classifiers and provided a comparative analysis based on an actual phishing email dataset proved that there is a significant improvement in accuracy in detecting phishing using data-driven methods. The results established that the essential role of effective data cleaning, normalisation, and feature extraction is important to increase the model performance. Although the conventional machine learning models, including Logistic Regression, SVM, and KNN, yielded decent baseline results, ensemble models, as well as deep learning models, demonstrated better performance in the detection. Specifically, the Random Forest model demonstrated the most balanced and confident result, which proves the significance of integrating contextual text interpretation with fire-engineered features in a specific phishing model.

This framework can be furthered in future studies by incorporating more sophisticated transformer-based processes to focus on more semantic connections within email texts. The mechanisms of deployment in real time and ongoing learning may be investigated so as to keep up with the changing strategy of phishing. Furthermore, it is suggested to expand the dataset and avoid the imbalance of the classes by means of advanced sampling methods, which would improve the generalisation and strength of the model.

It is also advisable that the organisation should embrace the implementation of hybrid machine learning and deep learning-based phishing systems as a means of enhancing email security. Although the introduction of sentiment analysis and the periodic retraining of the model may also use the technique of minimising false positives and enhance the effectiveness of the system in the long term.

REFERENCES

[1] Ahamad, G.N., Shafiullah, Fatima, H., Imdadullah, Zakariya, S.M., Abbas, M., Alqahtani, M.S. and Usman, M. (2023). Influence of Optimal Hyperparameters on the Performance of Machine Learning Algorithms for Predicting Heart Disease. *Processes*, 11(3), pp.1–28. doi:https://doi.org/10.3390/pr11030734.

[2] Al-Subaiey, A., Al-Thani, M., Alam, N.A., Antora, K.F., Khandakar, A. and Zaman, S.A.U. (2024). Novel interpretable and robust web-based AI platform for phishing email detection. *Computers & Electrical Engineering*, [online] 120, pp.1–23. doi:https://doi.org/10.1016/j.compeleceng.2024.109625.

[3] Conciatori, M., Valletta, A. and Segalini, A. (2024). Improving the quality evaluation process of machine learning algorithms applied to landslide time series analysis. *Computers & Geosciences*, 184, p.105531. doi:https://doi.org/10.1016/j.cageo.2024.105531.

[4] Frauenstein, E.D. and Flowerday, S. (2020). Susceptibility to phishing on social network sites: A personality information processing model. *Computers & Security*, 94, pp.1–18.

doi:https://doi.org/10.1016/j.cose.2020.101862.

[5] Kaggle (2024). *Phishing Email Dataset*. [online] Kaggle. Available at: https://www.kaggle.com/datasets/naserabdullahalam/phishing-email-dataset [Accessed 16 January 2026].

[6] Koukaras, P. and Tjortjis, C. (2025). Data Preprocessing and Feature Engineering for Data Mining: Techniques, Tools, and Best Practices. *AI*, 6(10), p.257. doi:https://doi.org/10.3390/ai6100257.

[7] Malashin, I., Tynchenko, V., Gantimurov, A., Nelyub, V. and Borodulin, A. (2024). Applications of Long Short-Term Memory (LSTM) Networks in Polymeric Sciences: A Review. *Polymers*, [online] 16(18), pp.2607–2607. doi:https://doi.org/10.3390/polym16182607.

[8] Marshall, N., Sturman, D. and Auton, J.C. (2023). Exploring the Evidence for Email Phishing Training: A Scoping Review. *Computers & Security*, [online] 139, p.103695. doi:https://doi.org/10.1016/j.cose.2023.103695.

[9] Naqvi, B., Perova, K., Farooq, A., Makhdoom, I., Oyedeji, S. and Porras, J. (2023). Mitigation Strategies against the Phishing Attacks: A Systematic Literature Review. *Computers & Security*, [online] 132. doi:https://doi.org/10.1016/j.cose.2023.103387.

[10] Onih, V.A. (2024). Phishing Detection Using Machine Learning: A Model Development and Integration. *International Journal of Scientific and Management Research*, [online] 07(04), pp.27–63. doi:https://doi.org/10.37502/ijsmr.2024.7403.

[11] Putra, E., Ubaidi, U., Zulfikri, A., Arifin, G. and Ilhamsyah, R.M. (2024). Analysis of Phishing Attack Trends, Impacts and Prevention Methods: Literature Study. *Brilliance Research of Artificial Intelligence*, 4(1), pp.413–421. doi:https://doi.org/10.47709/brilliance.v4i1.4357.

[12] Rupp, V. and von Grafenstein, M. (2024). Clarifying 'personal data' and the role of anonymisation in data protection law: Including and excluding data from the scope of the GDPR (more clearly) through refining the concept of data protection. *Computer Law & Security Review*, [online] 52, p.105932. doi:https://doi.org/10.1016/j.clsr.2023.105932.

[13] Yang, X., Yang, K., Cui, T., Chen, M. and He, L. (2022). A Study of Text Vectorization Method Combining Topic Model and Transfer Learning. *Processes*, [online] 10(2), p.350. doi:https://doi.org/10.3390/pr10020350.