Contents lists available at IdNNS

Indonesian Journal of Cyber-AI and Security Intelligence

Volume 1 ; Issue 1

journal homepage: https://journal.idnns.org/index.php/ijcasi

Comparative Approach in Evaluating the Impact of TF-IDF on Machine Learning Performance for Phishing Email Detection

Muhammad Azhar [a], Vinaye Armoogum[b], Sheeba Armoogum[b] and Hamudi Alamsyah[c]

[a] *Hong Kong Shue Yan University, Hong Kong, Hong Kong*
[b] *University of Technology, Mauritius, Port Louis, Mauritius*
[c] *School of Computer Science, Nusa Putra University, Sukabumi, Indonesia*

ARTICLE INFO

Keywords:
Phishing Emails
Classification
Random Forest
SVM
Logistic Regression
XGBoost
Natural Language Processing (NLP)
TF-IDF

ABSTRACT

Phishing emails continue to pose a critical threat to cybersecurity, causing substantial financial and data losses. To address this, accurate and efficient detection methods are essential. This study presents a comparative evaluation of four traditional machine learning algorithms Support Vector Machine (SVM), Random Forest, Logistic Regression, and XGBoost for phishing email detection, with a particular focus on the impact of Term Frequency-Inverse Document Frequency (TF-IDF) as a text feature extraction method. Experiments were conducted using a labeled dataset containing phishing and legitimate emails, evaluated under two scenarios: with and without the application of TF-IDF. Performance was assessed using accuracy, precision, recall, F1-Score, and AUC metrics. The results indicate that SVM, when combined with TF-IDF, achieves the highest classification performance, demonstrating its suitability for systems requiring high detection accuracy. Random Forest and XGBoost, while slightly behind in metrics, offer strong trade-offs between performance and computational efficiency, making them more suitable for deployment in resource-constrained environments. This study confirms that TF-IDF significantly enhances model performance and highlights the importance of feature engineering in traditional machine learning pipelines for phishing detection.

1. Introduction

The advancement of digital technology and communication has had a significant impact across various fields, from business to education. However, along with the benefits offered, increasingly complex cyber threats have also emerged, one of which is phishing attacks. Phishing is a form of cyber-attack that exploits psychological manipulation to deceive individuals into unknowingly providing sensitive information, such as passwords, financial data, or personal details. Attackers often impersonate trusted entities, such as banks or email services, through media like emails and instant messages.

In recent years, phishing attacks have become increasingly sophisticated and are one of the most commonly used methods of cyber-attacks. According to reports from various security agencies, phishing attacks are responsible for significant financial losses and damage to the reputation of companies. With the growing use of email and instant messaging applications as primary communication tools in the workplace, phishing attacks have become an increasingly urgent challenge to address (Alkhalil et al., 2021; Fette et al., 2007; Bergholz et al., 2010; Khonji et al., 2013; Xiang et al., 2011; Al Tawil et al., 2024; Kytidou et al., 2025; Gualberto et al., 2020; Narendra Kumar Reddy et al., 2020; Mourtaji et al., 2021; Shirazi et al., 2019).

Phishing attacks have evolved from simple email scams in the 1990s to more sophisticated multi-vector attacks using artificial intelligence and social media today. These attacks often disguise themselves as trusted entities to deceive individuals into revealing sensitive information, which can lead to identity theft, financial loss, or other types of fraud (Putra et al., 2024).

Traditional methods for detecting phishing often rely on rule-based approaches, such as domain blacklists or keyword scanning. Unfortunately, these approaches have limitations when faced with modern phishing attacks that are more complex and designed to evade detection. Therefore, machine learning and Natural Language Processing (NLP) technologies have emerged as more adaptive solutions. With machine learning, systems can learn from data and recognize new patterns that are not always easily identifiable by humans or rule-based systems. Meanwhile, NLP enables deeper text

Azhar@hksyu.edu (M.A.); varmoogum@utm.ac.mu (V. Armoogum); s.armoogum@uom.ac.mu (S. Armoogum); hamudi.alamsyah@nusaputra.ac.id (H. Alamsyah)
ORCID(s):

analysis to identify the context and linguistic patterns commonly used in phishing attacks.

Recent studies have demonstrated the effectiveness of combining machine learning and NLP (TF-IDF) in phishing detection. For instance, a research paper published in the SMU Data Science Review discusses the use of NLP techniques, clustering, and neural network-based machine learning models to identify phishing attempts, achieving accuracy scores exceeding 97% (Mittal and Sivaraman, 2022). Similarly, an article in IEEE Xplore presents an approach that utilizes NLP to analyze text and detect inappropriate statements indicative of phishing attacks (Peng et al., 2018).

These advancements highlight the potential of integrating machine learning and NLP to enhance phishing detection capabilities, offering more robust and adaptive solutions compared to traditional methods.

This research aims to compare the performance of traditional machine learning algorithms Random Forest, Support Vector Machine (SVM), Logistic Regression, and XGBoost in detecting phishing attacks on emails and web content using Natural Language Processing (NLP) techniques such as TF-IDF. By leveraging NLP, the system is expected to extract meaningful textual features and improve the accuracy of phishing detection. This comparative analysis will provide insights into the strengths and weaknesses of each algorithm, helping organizations choose the most effective solution for their cybersecurity needs. The proposed solution is not only a security tool but also a proactive step in safeguarding sensitive data and maintaining information integrity in an increasingly complex digital era.

2. Research Methodology

The research method used in this study is experimental, where a phishing email dataset is processed and analyzed using different machine learning models under controlled conditions. The experiment involves applying Natural Language Processing (NLP) techniques, particularly the TF-IDF method, to extract features from email text. Each machine learning model is then trained and tested under two conditions: with TF-IDF and without TF-IDF.

In a study conducted by da Rocha and Nogueira, an experimental method was used to investigate how remote learning through video-based experiments can impact students' understanding of physics concepts. Their research demonstrated that even in a non-traditional learning environment, measurable improvements in comprehension could be achieved when pre- and post-assessments were applied systematically. This concept aligns with the present study, where different machine learning models are evaluated under controlled settings with and without the TF-IDF feature. The purpose is to assess whether the application of natural language processing techniques, like TF-IDF, significantly contributes to improving the accuracy of phishing email detection (da Rocha and Nogueira, 2025). Zakiyah (2020) discusses the use of quasi-experimental designs, where participants are not randomly assigned but are still divided into distinct groups that receive different treatments. This approach is particularly useful in situations where full randomization is not feasible. Their method involves comparing outcomes between experimental and control groups to evaluate the effectiveness of an intervention. This is comparable to the present research, which applies TF-IDF as a treatment variable and examines its effect on various machine learning models by comparing their performance with and without the feature extraction technique (Zakiyah, 2020).

The results of both scenarios are compared using performance metrics to determine the effectiveness and consistency of each algorithm, and to evaluate the role of TF-IDF in improving phishing email detection.

2.1. Evaluation Method

In this study, several evaluation metrics are used to assess the performance of the developed model, namely:

Accuracy measures the percentage of correct predictions from the total test data.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Precision calculates the proportion of correctly predicted positive cases compared to all positive predictions made by the model.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

Recall (Sensitivity) measures how well the model detects actual phishing samples.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

F1-Score is the harmonic mean of precision and recall, providing a balanced evaluation of the model.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

2.2. Validation Method

The validation method used in this study is train-test split with an 80:20 ratio. The dataset is divided into 80% for training and 20% for testing, ensuring that the model is tested on unseen data to obtain a more realistic performance measurement.

Additionally, to enhance reliability, the experiment may also utilize k-fold cross-validation with k=10, which divides the data into 10 parts, trains the model on 9 parts, and tests it on the remaining part in a rotating manner.

2.3. Research Flow Diagram

Fig. 1 presents the overall research flow, from problem identification and literature study through data preprocessing, text vectorization, model development, evaluation, and real-world testing with .eml samples.

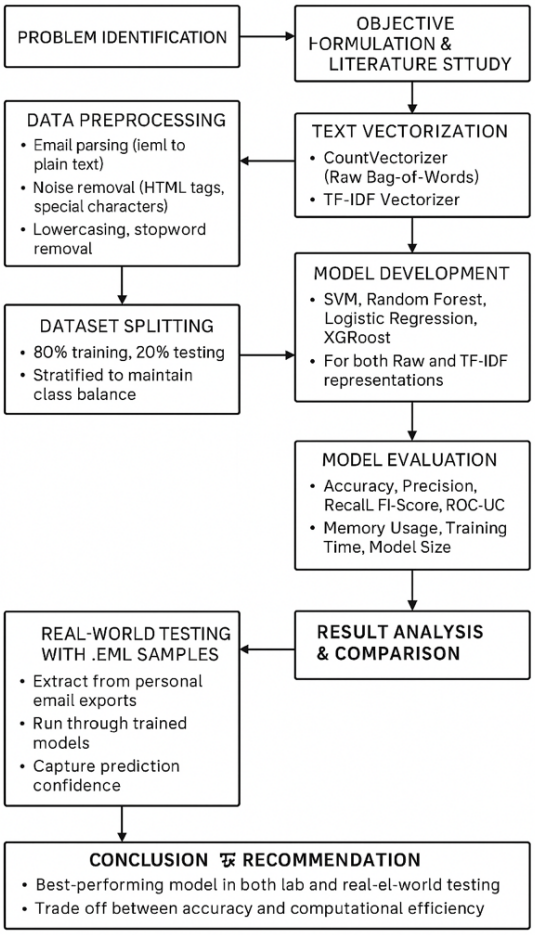

Figure 1: Research flow diagram

2.4. Data Analysis Methods

This study applies a quantitative and experimental approach using supervised machine learning techniques to analyze the data. The purpose is to compare the performance of four machine learning classifiers Random Forest, Support Vector Machine (SVM), Logistic Regression, and XGBoost in detecting phishing emails, both with and without the use of TF-IDF for feature extraction.

Two different approaches are used to transform raw email text into numeric features:

1. **TF-IDF (Term Frequency – Inverse Document Frequency).** TF-IDF assigns weights to each word based on how frequently it appears in an email (Term Frequency) and how rare it is across all emails (Inverse Document Frequency). The formula used is:

$$TF(t,d) = \frac{\text{Number of times word } t \text{ appears in document } d}{\text{Total words in document } d} \tag{5}$$

Inverse Document Frequency (IDF) measures how unique a word is across the entire set of documents.

$$IDF(t) = \log \frac{N}{DF(t)} \tag{6}$$

where N is the total number of documents in the corpus and $DF(t)$ is the number of documents containing word t. The final TF-IDF value is computed by multiplying the two components:

$$TF\text{-}IDF(t,d) = TF(t,d) \times IDF(t) \tag{7}$$

2. **Non-TF-IDF Baseline.** For comparison, simpler vectorization methods may be used (e.g., binary encoding or raw count), where no weighting is applied to text features.

2.5. Model Training

Fig. 2 shows the model training pipeline. The email dataset is split into training and testing sets (commonly 80:20). Each machine learning algorithm is trained separately under two scenarios: (1) with TF-IDF features, and (2) without TF-IDF features.

Each algorithm is trained on the training set using the hyperparameter grid summarized in Table 1. Each model learns to classify emails as either phishing (1) or non-phishing (0).

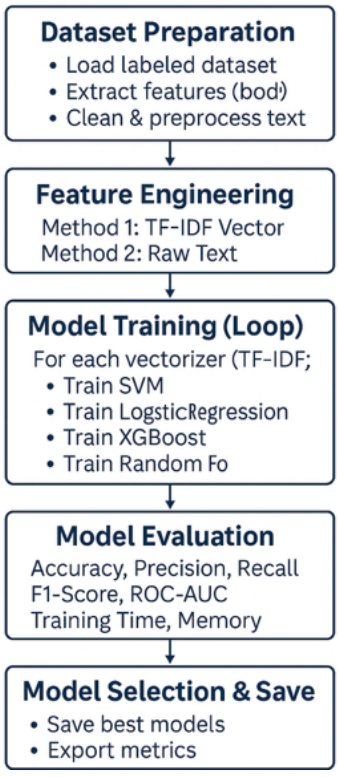

Figure 2: Model training pipeline

Table 1
Hyperparameter search space for each model

Model	Hyperparameter	Values to be Tested
Logistic Regression	C	[0.1, 1, 10]
	penalty	['l1', 'l2']
	solver	['liblinear']
Random Forest	n_estimators	[100, 200, 500]
	max_depth	[5, 10, 15]
	min_samples_split	[2, 5, 10]
SVM	C	[0.1, 1, 10]
	kernel	['linear', 'rbf']
	gamma	['scale', 'auto']
XGBoost	learning_rate	[0.01, 0.1, 0.2]
	n_estimators	[100, 200, 500]
	max_depth	[3, 5, 7]

2.6. Performance Evaluation

To assess and compare model performance, four commonly used evaluation metrics are applied: accuracy, precision, recall, and F1-Score, using the same definitions given in Section 2.1.

3. Research Results

This research aims to conduct a comparative analysis of four machine learning algorithms Random Forest, Support Vector Machine (SVM), Logistic Regression, and XGBoost for detecting phishing emails. The training process was carried out using two types of text representations: raw representation using CountVectorizer and weighted representation using Term Frequency–Inverse Document Frequency (TF-IDF) via TfidfVectorizer. The dataset used was CEAS 2008, labeled binarily as 1 for phishing and 0 for non-phishing.

The training results were evaluated using several metrics, including accuracy, precision, recall, F1-Score, ROC-AUC, training time, memory usage, and model size. All results were presented in tables comparing models across both raw and TF-IDF representations. To aid understanding, performance comparison charts were also provided. The comparison chart indicates that all models experienced an improvement in accuracy after the application of TF-IDF. The most significant increases were observed in SVM and Random Forest, suggesting that TF-IDF-based feature representation strengthens the model's ability to recognize word patterns that distinguish phishing emails from legitimate ones.

TF-IDF also positively impacted precision across all models. SVM achieved the highest precision after using TF-IDF, indicating an improved ability to reduce false positives, or instances where legitimate emails are incorrectly classified as phishing.

For the recall metric, consistent improvement was also observed, particularly in XGBoost and Logistic Regression. This implies that TF-IDF helps the models become more sensitive to actual phishing emails, thereby reducing the number of false negatives.

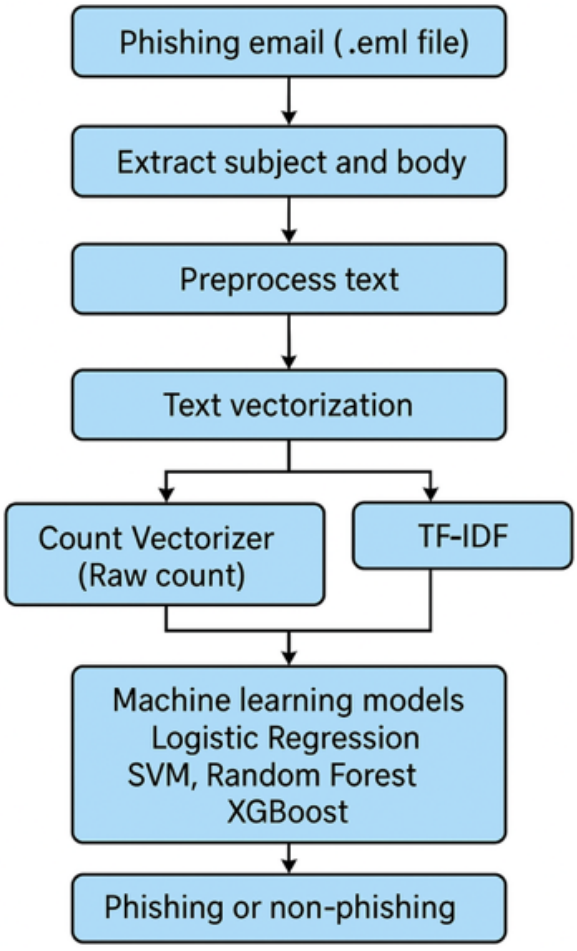

Figure 3: Text vectorization and classification flow

As the F1-Score represents a harmonic mean of precision and recall, the graph shows a balanced and significant increase in this metric for all models after implementing TF-IDF. SVM and Random Forest again recorded the highest F1-Scores, confirming their strong performance in both precision and sensitivity.

All models exhibited improved AUC values after applying TF-IDF, further reinforcing that richer and more structured feature representation helps enhance the model's discriminative ability between phishing and legitimate emails.

Although TF-IDF improved performance, the graph reveals that training time increased, particularly for the SVM model. This can be attributed to the fact that TF-IDF generates a significantly larger number of features compared to simpler representations like bag-of-words.

The comparison also shows a rise in memory usage and model size after applying TF-IDF, with the most noticeable impact on Random Forest (model size) and SVM (memory usage). This highlights the trade-off between accuracy and computational efficiency that must be considered in real-world deployment scenarios. These visualizations offer a clear view of the strengths and weaknesses of each model on the two types of text representation.

The phishing email detection process in this study is systematically designed by considering essential stages related to text data processing and the application of machine learning algorithms. The primary goal of the system is to classify whether an email belongs to the phishing or non-phishing category based on the textual content contained within.

The system operates by receiving input in the form of .eml files, a standard format used to store complete email data, including headers, subject lines, body text, and attachments. The process begins by extracting relevant parts, namely the subject and body, which contain the core message of the email.

Once extracted, the next stage is preprocessing. This step involves removing HTML tags, special characters, numbers, URLs, and non-informative words such as stopwords. Additionally, all text is converted to lowercase and stemmed using a stemming algorithm to reduce word variation.

After preprocessing, the clean text is converted into numerical representations using two approaches raw count vectorization and TF-IDF weighting. These representations serve as feature inputs for the machine learning algorithms used for model training and prediction.

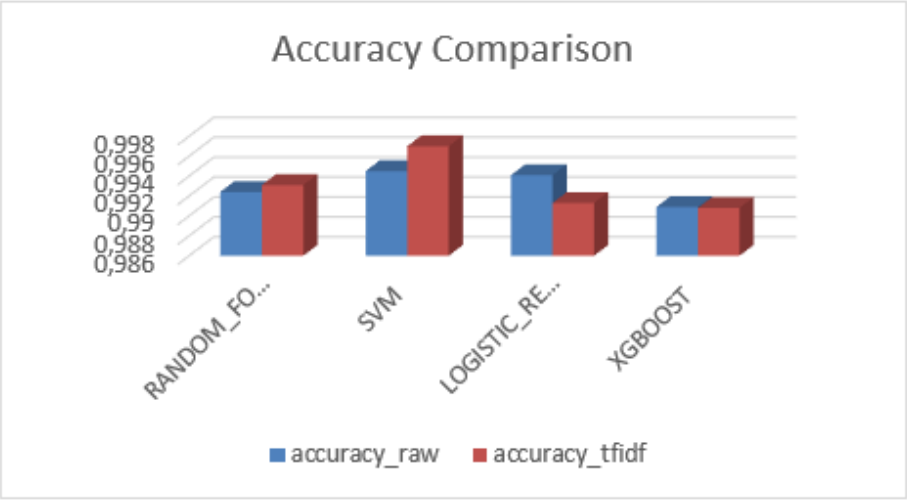

Figure 4: Accuracy comparison between raw and TF-IDF representations

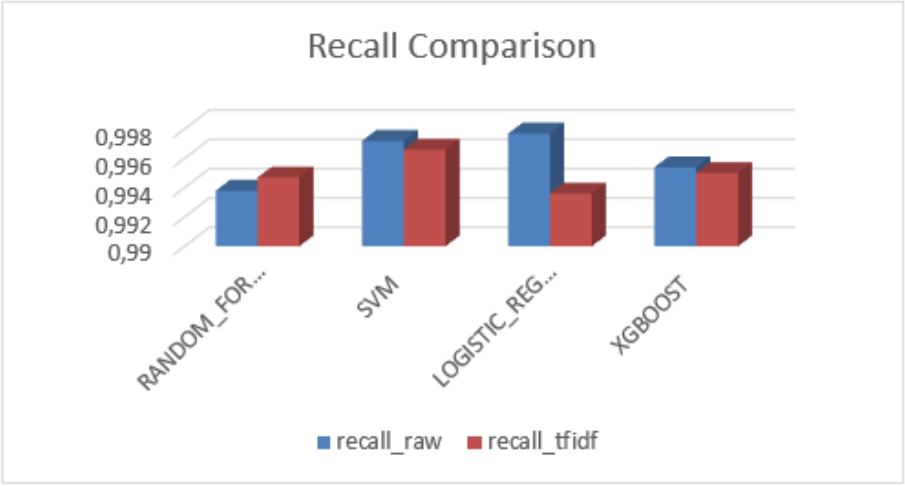

Figure 5: Recall comparison between raw and TF-IDF representations

Four machine learning models are employed in this system Logistic Regression, Support Vector Machine (SVM), Random Forest, and XGBoost. These models were selected based on their strengths in handling text data and their effectiveness in producing reliable classification results. Each model is trained using both feature representations (raw count and TF-IDF), allowing for a comparative evaluation of the impact of feature representation on model performance.

The training process is conducted outside of the system's runtime to avoid computational overhead during usage. Trained models are serialized using the joblib library and later reloaded when the system is run for new email predictions.

With this approach, the phishing email detection system is able to operate efficiently, automatically identify threats based on textual content, and be deployable in environments with limited resources such as cloud systems or local devices.

One of the most critical aspects of text-based classification systems is how the text data is represented numerically for processing by machine learning algorithms. This representation, known as text vectorization or textual feature extraction, significantly affects the model's ability to identify patterns specific to phishing emails, as illustrated in Fig. 3. This study explores two main feature representation approaches:

1. **Count Vectorizer (Raw Count).** This method represents each document as a vector based on the frequency of each word in the corpus. While this approach is simple and effective in many contexts, it does not consider the importance of a word across all documents. Common words like "click," "account," or "login" may appear frequently but do not always indicate phishing. As a result, this method can produce models that rely too heavily on frequency rather than contextual meaning.

Table 2
Model performance with raw (CountVectorizer) representation

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC	Training Time (s)	Model Size (KB)
Random Forest	0.9924	0.9927	0.9938	0.9932	0.9994	69.7389	42333.431
SVM	0.9945	0.9929	0.9972	0.9951	0.9992	246.8244	938.5576
Logistic Regression	0.9941	0.9918	0.9977	0.9947	0.9995	0.4432	1109.8584
XGBoost	0.9909	0.9884	0.9954	0.9919	0.9992	12.1793	157.4834

2. **TF-IDF (Term Frequency – Inverse Document Frequency).** This method assigns a weight to each word based on its frequency in a document and its rarity across the entire corpus. Words that appear frequently in one document but rarely elsewhere are considered more informative. In phishing detection, TF-IDF is effective at highlighting distinctive terms that commonly appear in phishing emails but are rare in legitimate ones, such as "account verification," "temporary suspension," or "click this link."

Experiments in this study demonstrate that using TF-IDF as a feature representation yields better model performance compared to raw count. This is because TF-IDF reduces the influence of common, less informative words while emphasizing terms that are more indicative of phishing attacks.

This performance difference is validated through improvements in precision, recall, and F1-score across all tested models. TF-IDF also enhances the model's ability to differentiate between phishing and legitimate emails, even when the message is crafted to resemble official communication.

3.1. Model Performance with Raw Representation

Based on evaluation results, SVM using raw representation yielded the highest accuracy of 0.9945, followed by Logistic Regression and Random Forest. However, in terms of F1-Score, Random Forest achieved the highest value of 0.9932. This indicates that Random Forest maintained a better balance between precision and recall on raw data.

All models had ROC-AUC scores close to 1.000, indicating excellent classification capabilities. However, it is worth noting that SVM's training time was very high (246.82 seconds), significantly higher than other models, indicating that SVM is less computationally efficient under this representation. Table 2 summarizes the detailed results for the raw representation.

3.2. Model Performance with TF-IDF Representation

Using TF-IDF representation led to a significant increase in the F1-Score of the SVM model (0.9974), making it the overall best-performing model. Logistic Regression and XGBoost also yielded high F1-Scores above 0.99 but did not outperform SVM. Random Forest, previously superior with raw data, showed a slight F1-Score increase to 0.9938 with TF-IDF.

However, SVM again exhibited a very long training time on TF-IDF (1062.26 seconds), indicating high resource requirements. XGBoost presented itself as a favorable alternative with a high F1-Score (0.9918) and a much faster training time. Table 3 summarizes the detailed results for the TF-IDF representation.

3.3. Final Comparison of Raw vs. TF-IDF Representation

Each model demonstrated different performance depending on the representation used. The final metric comparison between both representations (Tables 2 and 3) showed that TF-IDF yielded higher F1-Scores, precision, and recall in most models. SVM and Random Forest achieved higher F1-Scores using TF-IDF compared to raw vectors. This proves that TF-IDF representation is more effective in capturing important textual information in phishing emails than CountVectorizer.

Logistic Regression and XGBoost did not show significant improvements and even had slight declines, though the difference was minor and still within high-performance margins. This suggests that while TF-IDF may not always be better for every model, it generally delivers more stable and high-quality results for phishing detection.

The main reason why TF-IDF performs better is its ability to reduce the weight of common words (like "the," "is") and emphasize more unique and informative words in the phishing context. This helps models detect phishing patterns more accurately.

3.4. Visualization and Source Metric Values

The evaluation metrics in this study were derived from testing on 20% of the CEAS 2008 dataset. Evaluations were performed using the `accuracy_score`, `precision_score`, `recall_score`, `f1_score`, and `roc_auc_score` functions from the scikit-learn library (Pedregosa et al.). Model sizes were measured from .pkl files, and memory consumption was measured using the psutil module. Visualizations of F1-Score, training time, and model size were shown using bar charts to clearly depict the performance differences among models.

The system flow diagram of training and evaluation is also included to show how the dataset was processed, split, vectorized, and paired with each algorithm before being evaluated. Additionally, Tables 2 and 3 present the F1-Score and accuracy comparisons between TF-IDF and raw representations across all models.

Table 3
Model performance with TF-IDF representation

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC	Training Time (s)	Model Size (KB)
Random Forest	0.9931	0.9929	0.9947	0.9938	0.9994	65.1513	36033.899
SVM	0.9970	0.9982	0.9966	0.9974	0.9998	1062.2611	2849.3232
Logistic Regression	0.9913	0.9909	0.9936	0.9922	0.9994	0.7965	1109.8584
XGBoost	0.9908	0.9886	0.9950	0.9918	0.9992	47.1200	154.7607

3.5. Experimentation Using EML Format

To assess the performance and real-world applicability of the phishing detection system, a practical experiment was conducted using .eml files email messages in their native format. This test aims to simulate realistic conditions in which the system will operate by analyzing the capability of trained machine learning models to process and classify incoming emails. Figs. **??**–6 present the prediction results, vectorizer, model, and confidence score obtained for each of the 17 tested .eml samples.

To further assess model robustness in real-world scenarios, a series of tests were conducted on ten .eml files manually labeled as non-phishing. These files originated from actual personal email exports, thereby simulating realistic deployment conditions. The evaluation involved comparing model predictions against the known labels, using both raw CountVectorizer and TF-IDF representations across four machine learning models: Support Vector Machine (SVM), Random Forest, Logistic Regression, and XGBoost.

Among all models tested, SVM combined with TF-IDF representation demonstrated the most consistent and stable performance. It correctly predicted 9 out of 10 emails as non-phishing, aligning with the true labels. Notably, even when its confidence scores were lower than other models such as the raw SVM it still delivered accurate predictions. This suggests that a model's confidence score is not always a reliable indicator of classification correctness, especially when dealing with unstructured and ambiguous data such as email content. In contrast, models like XGBoost and Logistic Regression with TF-IDF occasionally misclassified legitimate emails as phishing, despite showing high confidence levels. These outcomes highlight a tendency towards overconfidence, likely due to overfitting on certain features that coincidentally appear in both phishing and legitimate messages.

TF-IDF proved beneficial overall in enhancing model sensitivity to discriminative terms by down-weighting common words and emphasizing rare but relevant tokens. However, its impact varied depending on the algorithm. While SVM leveraged TF-IDF's feature weighting effectively, boosting its precision and resilience, models like XGBoost and Logistic Regression appeared more susceptible to noise introduced by the expanded feature space. This resulted in a higher false-positive rate, particularly in emails with content that superficially resembled phishing attempts.

Interestingly, the raw (non-TF-IDF) models frequently produced higher confidence values, yet this did not consistently translate to better accuracy. For instance, several models with raw input confidently predicted phishing when the emails were, in fact, legitimate. This reinforces the importance of evaluating models not solely on confidence but on their real-world classification consistency. SVM with TF-IDF stood out by maintaining a balance between precision and generalization, demonstrating that it can reliably detect legitimate emails without being misled by outlier terms or content irregularities.

In summary, the empirical testing on unlabeled .eml files highlights that SVM with TF-IDF provides the most reliable and consistent classification performance under real-world conditions. Despite occasionally yielding modest confidence levels, its predictions were more often aligned with the true labels. This makes it a strong candidate for practical phishing detection systems, where the ability to avoid false alarms is just as critical as identifying true threats. Ultimately, the findings underscore that effective phishing detection relies not only on high-confidence scores but also on a model's robustness, interpretability, and adaptability to varied email structures.

4. Conclusions

4.1. Conclusions and Recommendations

This study was conducted to evaluate the effectiveness of traditional machine learning algorithms in detecting phishing emails through the application of Natural Language Processing (NLP) techniques. The research specifically compared the performance of four established algorithms: Random Forest, Support Vector Machine (SVM), Logistic Regression, and XGBoost. Two text feature extraction methods raw text representation (using CountVectorizer) and TF-IDF (Term Frequency–Inverse Document Frequency) were applied to investigate their influence on model performance.

The dataset used was the CEAS 2008 corpus, consisting of labeled phishing and legitimate (ham) emails. The experimental workflow included data preprocessing, text vectorization, model training, and evaluation using key performance metrics: accuracy, precision, recall, F1-Score, ROC-AUC, training time, memory consumption, and model size.

The results demonstrate that the application of TF-IDF significantly improved model performance across most algorithms, with SVM achieving the highest accuracy and

File-wise Evaluation (Based on Label)

File	Vectorizer	Model	True Label	Prediction	Correct?	Confidence
email20.eml	RAW	RANDOM_FOREST	Non-Phishing	Phishing	No	0.72
email20.eml	RAW	SVM	Non-Phishing	Phishing	No	0.85
email20.eml	RAW	LOGISTIC_REGRESSION	Non-Phishing	Phishing	No	0.90
email20.eml	RAW	XGBOOST	Non-Phishing	Phishing	No	0.84
email20.eml	TFIDF	RANDOM_FOREST	Non-Phishing	Phishing	No	0.58
email20.eml	TFIDF	SVM	Non-Phishing	Non-Phishing	Yes	0.88
email20.eml	TFIDF	LOGISTIC_REGRESSION	Non-Phishing	Phishing	No	0.52
email20.eml	TFIDF	XGBOOST	Non-Phishing	Phishing	No	0.85

Figure 6: Prediction result for test email sample

F1-Score when paired with TF-IDF. Additionally, Random Forest and XGBoost offered strong trade-offs between predictive performance and computational efficiency, making them suitable for resource-constrained deployments.

4.2. Recommendations

Based on the findings of this research, several recommendations and directions for future work can be proposed to further enhance the phishing email detection system and similar text-classification systems. It is essential to investigate additional strategies to enhance model accuracy and efficiency. The following recommendations are proposed:

- The classifiers can be further optimized by tuning hyperparameters more extensively, using approaches such as Grid Search, Randomized Search, Bayesian Optimization, and similar techniques.
- Expand the dataset to include more email samples and additional phishing patterns, which may enhance the adaptability and generalization of the models.
- Consider exploring hybrid frameworks that combine multiple machine learning or deep learning techniques, such as ensembles of Gradient Boosting Machines and XGBoost.
- Integrate the system into a live email platform to provide insights into how well the models adapt to evolving phishing patterns and handle real-time email traffic.

These recommendations highlight the importance of continuous refinement and exploration in phishing detection research. By addressing the current limitations and leveraging alternative approaches, the system's capabilities can be enhanced to provide more robust, efficient, and reliable phishing detection solutions. These efforts will contribute significantly to securing digital communication systems against the evolving threat of phishing.

References

Al Tawil, A., Almazaydeh, L., Qawasmeh, D., Qawasmeh, B., Alshinwan, M., Elleithy, K., 2024. Comparative analysis of machine learning algorithms for email phishing detection using tf-idf, word2vec, and bert. Computers, Materials & Continua 81, 3395–3412. doi:10.32604/cmc.2024.057279.

Alkhalil, Z., Hewage, C., Nawaf, L., Khan, I., 2021. Phishing attacks: A recent comprehensive study and a new anatomy. Frontiers in Computer Science 3, 1–23. doi:10.3389/fcomp.2021.563060.

Bergholz, A., De Beer, J., Glahn, S., Moens, M.F., Paaß, G., Strobel, S., 2010. New filtering approaches for phishing email. Journal of Computer Security 18, 7–35. doi:10.3233/JCS-2010-0371.

da Rocha, C.A., Nogueira, M.S., 2025. Active learning methodology applied to a remote projectile launch experiment: Students first impressions. Physics Education , 1–20.

Fette, I., Sadeh, N.M., Tomasic, A., 2007. Learning to detect phishing emails, in: Proceedings of the 16th International Conference on World Wide Web, ACM. pp. 649–656. doi:10.1145/1242572.1242660.

Gualberto, E.S., De Sousa, R.T., De Brito Vieira, T.P., Da Costa, J.P.C.L., Duque, C.G., 2020. The answer is in the text: Multi-stage methods for phishing detection based on feature engineering. IEEE Access 8, 223529–223547. doi:10.1109/ACCESS.2020.3043396.

Khonji, M., Iraqi, A., Jones, A., 2013. Phishing detection: A literature survey. IEEE Communications Surveys & Tutorials 15, 2091–2121. doi:10.1109/SURV.2013.032213.00009.

Kytidou, E., Tsikriki, T., Drosatos, G., Rantos, K., 2025. Machine learning techniques for phishing detection: A review of methods, challenges, and future directions. Intelligent Decision Technologies 19, 4356–4379. doi:10.1177/18724981251366763.

Mittal, A., Sivaraman, R., 2022. Phishing detection using natural language processing and machine learning. SMU Data Science Review 6.

Mourtaji, Y., Bouhorma, M., Alghazzawi, D., Aldabbagh, G., Alghamdi, A., 2021. Hybrid rule-based solution for phishing url detection using convolutional neural network. Wireless Communications and Mobile Computing 2021. doi:10.1155/2021/8241104.

Narendra Kumar Reddy, G., Sabarimalai Manikandan, M., Narasimha Murty, N.V.L., 2020. On-device integrated ppg quality assessment and sensor disconnection/saturation detection system for iot health monitoring. IEEE Transactions on Instrumentation and Measurement 69, 6351–6361. doi:10.1109/TIM.2020.2971132.

Pedregosa, F., Giraud-Carrier, C., Michel, V., Thirion, B., Grisel, O., Duchesnay, É., . Scikit-learn: Machine learning in python. https://scikit-learn.org.

Peng, T., Harris, I., Sawa, Y., 2018. Detecting phishing attacks using natural language processing and machine learning, in: 2018 IEEE 12th International Conference on Semantic Computing (ICSC), pp. 300–301. doi:10.1109/ICSC.2018.00056.

Putra, F.P.E., Ubaidi, U., Zulfikri, A., Arifin, G., Ilhamsyah, R.M., 2024. Analysis of phishing attack trends, impacts and prevention methods: Literature study. Brilliance: Research of Artificial Intelligence 4, 413–421. doi:10.47709/brilliance.v4i1.4357.

Shirazi, H., Bezawada, B., Ray, I., Anderson, C., 2019. Adversarial sampling attacks against phishing detection. Lecture Notes in Computer Science 11559, 83–101. doi:10.1007/978-3-030-22479-0_5.

Xiang, G., Hong, J., Rose, C.P., Cranor, L.F., 2011. Cantina+: A feature-rich machine learning framework for detecting phishing web sites. ACM Transactions on Information and System Security 14, 21:1–21:28. doi:10.1145/2019599.2019606.

Zakiyah, S., 2020. Metodologi penelitian quasi eksperimen. Journal of Education 5, 183–192.