IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 7, July 2026

DOI: 10.17148/IJARCCE.2026.15746

AI-Based Phishing Email Detection Using NLP and Classical Machine Learning Classifiers

Dr.Harsha Vikas Patil[1], Mr.Sai Shrikrishna Patil[2]

Assistant Professor, Department of Computer Application, MAEER's MIT Arts, Commerce and Science College, Alandi (D), Pune, India[1]

Student, Computer Engineering (Cyber Security), D.Y. Patil International University, Akurdi, Pune, India[2]

Abstract: Phishing email remains one of the most persistent and cost-effective attack vectors used to compromise credentials, deliver malware, and defraud individuals and organizations. This study presents an end-to-end machine learning pipeline for automated phishing email detection using a large, publicly sourced corpus of 82,486 labeled emails (39,595 legitimate and 42,891 phishing). Raw email text was cleaned through lowercasing, URL removal, non-alphabetic character stripping, and stopword elimination, and transformed into a 5,000-dimensional Term Frequency-Inverse Document Frequency (TF-IDF) feature space. Three supervised classifiers Multinomial Naïve Bayes, Linear Support Vector Machine (SVM), and Random Forest were trained on an 80:20 stratified train-test split and evaluated using accuracy, precision, recall, and F1-score. The Linear SVM achieved the strongest overall performance (accuracy = 98.33%, precision = 98.46%, recall = 98.33%, F1-score = 98.40%), outperforming Naïve Bayes (accuracy = 95.93%) and Random Forest (accuracy = 94.32%). The trained SVM model and TF-IDF vectorizer were serialized for deployment, and a demonstration function correctly classified an unseen, realistic phishing message. The findings confirm that a lightweight, interpretable TF-IDF plus linear-classifier pipeline can achieve detection performance competitive with far more computationally expensive deep learning and transformer-based approaches reported in recent literature, making it a practical candidate for resource-constrained or real-time email security deployments.

Keywords: Phishing Detection, Machine Learning, Natural Language Processing, TF-IDF, Support Vector Machine, Naïve Bayes, Random Forest, Cybersecurity, Text Classification, Email Security.

I. INTRODUCTION

Electronic mail remains the single most widely exploited channel for social engineering attacks despite decades of advances in spam filtering and network-level security controls. Phishing emails impersonate trusted senders banks, employers, government agencies, or well-known brands to induce recipients to disclose credentials, transfer funds, or execute malicious attachments. The scale of the problem continues to grow: the Anti-Phishing Working Group recorded roughly one million phishing attacks per quarter through 2025 and into early 2026, with attacks against telecom, SaaS/webmail, and financial-services sectors rising sharply as attackers diversify beyond simple email lures into SMS, QR-code, and business-email-compromise (BEC) vectors [1].

Traditional defenses such as blacklists, signature-based filters, and hand-crafted heuristic rules struggle to keep pace with the speed at which phishing campaigns mutate their wording, sender domains, and social engineering pretexts. This limitation has motivated three decades of research into applying statistical and, more recently, deep learning methods to the problem of automatically distinguishing phishing emails from legitimate correspondence based on their textual content. Classical machine learning pipelines that combine a bag-of-words or TF-IDF representation of email text with a supervised classifier most commonly Naïve Bayes, Support Vector Machines (SVM), or ensemble tree methods such as Random Forest remain attractive because they are computationally inexpensive to train and deploy, are relatively transparent compared with deep neural architectures, and have repeatedly demonstrated accuracy in the mid-to-high nineties on benchmark corpora [2].

This paper reports the design, implementation, and evaluation of such a classical machine learning pipeline for phishing email detection. Using a merged, publicly sourced dataset of 82,486 emails, the study builds a complete natural language processing (NLP) workflow text cleaning, stopword removal, TF-IDF vectorization, model training, and quantitative evaluation and benchmarks three widely used classifiers: Multinomial Naïve Bayes, Linear SVM, and Random Forest. The trained model is further packaged for practical deployment, and its behavior is illustrated on an unseen, realistic phishing message. The specific objectives of the study are to: (i) build a reproducible preprocessing and feature-extraction pipeline suited to noisy, real-world email text; (ii) compare the discriminative performance of three canonical machine learning classifiers under identical experimental conditions; (iii) identify the classifier offering the best accuracy-to-complexity trade-off for lightweight deployment; and (iv) situate the empirical results within the wider

DOI: 10.17148/IJARCCE.2026.15746

body of recent phishing-detection literature, including deep learning and transformer-based approaches, to clarify where classical methods remain competitive and where they fall short.

II. LITERATURE REVIEW

A. Classical Machine Learning Approaches

A substantial body of work has evaluated Naïve Bayes, SVM, and Random Forest classifiers for phishing and spam email detection using TF-IDF or bag-of-words feature representations. In a comparative study across ten datasets and fourteen ML/DL models, Alhuzali et al. [2] reported that an SVM classifier trained on a merged dataset of approximately 82,500 emails combining six public sources achieved an accuracy of 99.19%, closely matching the scale and composition of the dataset used in the present study; the authors nonetheless found that deep learning architectures outperformed classical ML approaches by an average margin of roughly 4.7% across their full benchmark. Similarly, comparative evaluations of Naïve Bayes, Logistic Regression, and SVM on fraudulent-email corpora have consistently found SVM to be the strongest of the three, while Naïve Bayes' independence assumption is reported to degrade performance when correlated lexical cues such as URLs and HTML markers co-occur in phishing text. TF-IDF weighting optimized with multi-metric criteria has also been shown to substantially improve Naïve Bayes accuracy on spam corpora, underscoring the sensitivity of classical pipelines to preprocessing and feature-engineering choices [3].

B. Deep Learning and Transformer-Based Approaches

More recent studies have shifted toward deep neural architectures. Altwaijry et al. [4] proposed a one-dimensional convolutional model (1D-CNNPD) augmented with recurrent layers (LSTM, BiLSTM, GRU, BiGRU) and reported that the best-performing variant achieved 99.68% accuracy and 100% precision on benchmark phishing corpora, illustrating the ceiling that deep architectures can reach when sufficient training data is available. Transformer-based models have pushed performance further still: fine-tuned BERT and RoBERTa models have been reported to reach accuracies in the 94–99% range with strong robustness to adversarial paraphrasing and disguised phishing attempts, and lightweight transformer variants such as DistilBERT and CatBERT have been proposed specifically to reduce inference cost while retaining most of BERT's accuracy advantage [5]. These findings collectively establish transformer architectures as the current performance ceiling for text-based phishing detection, but at a substantially higher computational and infrastructural cost than classical TF-IDF pipelines.

C. The Cost-Accuracy Trade-off

A recurring theme across this literature is the trade-off between marginal accuracy gains from deep and transformer-based models and the training time, memory footprint, and interpretability advantages of classical ML pipelines. Because TF-IDF plus linear-classifier pipelines require no GPU infrastructure, train in minutes rather than hours, and produce feature weights that are directly inspectable, they remain attractive for real-time filtering at the mail-gateway level, for resource-constrained deployments, and for settings where model transparency is a regulatory or trust requirement. The present study contributes to this literature by providing a controlled, same-dataset comparison of three classical classifiers, offering an empirical basis for choosing among them when deep learning is not a practical option.

III. RESEARCH GAP AND MOTIVATION

While numerous studies report high accuracy for both classical and deep learning phishing detectors, three gaps motivate the present work. First, many published comparisons train each classifier on different datasets or preprocessing pipelines, making like-for-like comparison difficult; the present study holds the dataset, cleaning pipeline, and feature space constant across all three classifiers. Second, relatively few studies report a complete, reproducible pipeline from raw text to a serialized, deployable model rather than reporting only headline accuracy figures; this study documents every pipeline stage and delivers a saved model artifact together with a working prediction function. Third, given the accuracy-cost trade-off identified in Section II-C, there is a continuing need for empirical evidence on how far classical, low-cost pipelines can go before deep learning becomes strictly necessary, particularly for practitioners and institutions without access to GPU infrastructure.

IV. METHODOLOGY

A. Dataset Description

The study uses a labeled email corpus (phishing_email.csv) comprising 82,486 records across two columns: text_combined (the raw email text) and label (0 = legitimate, 1 = phishing). The class distribution is close to balanced, with 42,891 phishing emails (52.0%) and 39,595 legitimate emails (48.0%), a composition and scale consistent with merged public phishing corpora reported elsewhere in the literature [2]. No missing values were present in either column; after removing duplicate records, 82,078 unique emails remained available for preprocessing.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 7, July 2026

DOI: 10.17148/IJARCCE.2026.15746

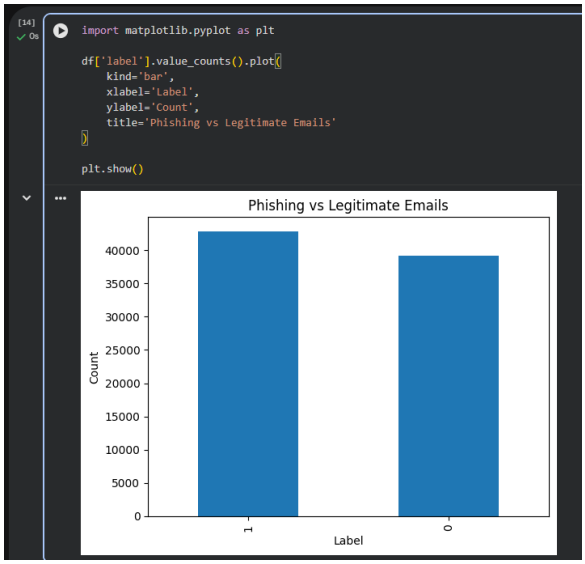

Fig. 1. Distribution of phishing (label = 1) versus legitimate (label = 0) emails in the dataset.

B. Data Preprocessing

Raw email text was cleaned using a deterministic function that (i) lower-cased all text, (ii) removed URLs using a regular-expression match on http-prefixed substrings, (iii) stripped all non-alphabetic characters (digits and punctuation), (iv) tokenized the remaining text on whitespace, and (v) removed English stopwords using the NLTK stopword corpus. The cleaned tokens were re-joined into a single normalized string per email, producing a clean_text field used for all downstream feature extraction. This preprocessing choice trades some information loss (e.g., numeric values, punctuation-based urgency cues such as exclamation marks) for a smaller, less sparse vocabulary that is easier for linear and probabilistic classifiers to model.

C. Feature Extraction: TF-IDF Vectorization

The cleaned corpus was transformed into a numerical feature matrix using scikit-learn's TfidfVectorizer, restricted to the 5,000 most informative terms by document frequency (max_features = 5000). Term Frequency-Inverse Document Frequency (TF-IDF) weighting was chosen over a raw bag-of-words representation because it down-weights terms that occur ubiquitously across the corpus (e.g., generic email boilerplate) while up-weighting terms that are discriminative of a particular class, a property repeatedly shown to benefit both Naïve Bayes and SVM classifiers on email-classification tasks [3]. The resulting sparse feature matrix had dimensions of 82,486 × 5,000.

D. Train-Test Split

The feature matrix and label vector were partitioned into training (80%) and testing (20%) subsets using stratified random sampling (random_state = 42) to preserve the original class ratio in both partitions. This yielded 65,988 training examples and 16,498 held-out test examples, with the test set comprising 7,919 legitimate and 8,579 phishing emails.

E. Classification Algorithms

Three supervised classifiers, each representing a distinct modeling paradigm, were trained on the identical TF-IDF feature space:

Multinomial Naïve Bayes. A probabilistic classifier that assumes conditional independence between features given the class label. It is computationally efficient and a long-standing baseline for text classification, though its independence assumption can be violated by correlated lexical patterns typical of phishing text (e.g., co-occurring URL tokens and urgency language).

Linear Support Vector Machine (LinearSVC). A maximum-margin linear classifier that seeks the hyperplane best separating the two classes in the high-dimensional TF-IDF space. SVMs are well suited to sparse, high-dimensional text data and have consistently ranked among the top-performing classical models in the phishing-detection literature reviewed in Section II.

Random Forest. An ensemble of decision trees (n_estimators = 50, max_depth = 20) trained on bootstrapped samples and random feature subsets, then aggregated by majority vote. Random Forest was included as a non-linear, ensemble baseline to test whether tree-based partitioning of the TF-IDF space could match the linear decision boundary learned by the SVM.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 7, July 2026

DOI: 10.17148/IJARCCE.2026.15746

F. Evaluation Metrics

Model performance was assessed on the held-out test set using four standard classification metrics: accuracy (overall proportion of correct predictions), precision (proportion of predicted-phishing emails that were truly phishing), recall (proportion of actual phishing emails correctly identified), and F1-score (harmonic mean of precision and recall). In a security-critical application such as phishing detection, recall is of particular practical importance because a missed phishing email (false negative) exposes the recipient to direct harm, whereas a legitimate email incorrectly flagged as phishing (false positive) is a comparatively lower-cost inconvenience. Confusion matrices were also generated for each classifier to visualize the distribution of true positives, true negatives, false positives, and false negatives.

V. RESULTS AND ANALYSIS

Table I summarizes the performance of the three classifiers on the 16,498-email held-out test set.

TABLE I COMPARATIVE PERFORMANCE OF NAÏVE BAYES, SVM, AND RANDOM FOREST

Model	Accuracy	Precision	Recall	F1-Score
Multinomial Naïve Bayes	0.9593	0.9768	0.9442	0.9602
Linear SVM	0.9833	0.9846	0.9833	0.9840
Random Forest	0.9432	0.9165	0.9801	0.9472

The Linear SVM achieved the best performance on every metric, correctly classifying 98.33% of test emails with a precision of 98.46% and a recall of 98.33%, yielding an F1-score of 0.9840. Multinomial Naïve Bayes was the second-best model overall (accuracy = 95.93%, F1 = 0.9602); notably, its precision (0.9768) exceeded its recall (0.9442), indicating that when Naïve Bayes flagged an email as phishing it was usually correct, but it missed a somewhat larger share of actual phishing emails than the SVM did. Random Forest recorded the lowest accuracy of the three models (94.32%) despite achieving the highest raw recall (0.9801), reflecting a precision-recall trade-off in which the ensemble was aggressive at flagging phishing emails (precision = 0.9165) at the cost of a higher false-positive rate on legitimate mail.

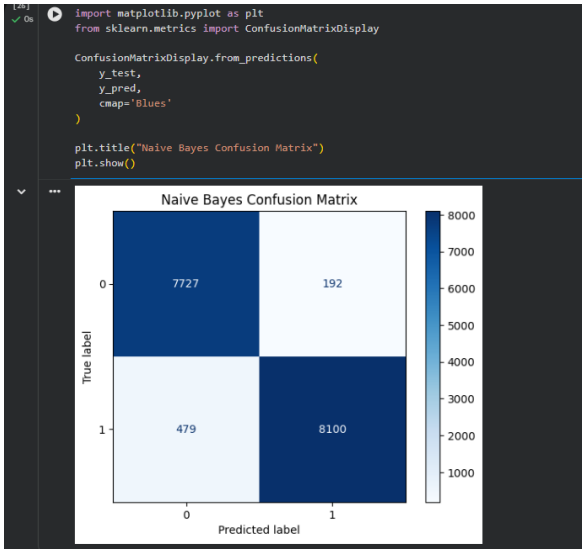

Fig. 2. Confusion matrix Multinomial Naïve Bayes.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 7, July 2026

DOI: 10.17148/IJARCCE.2026.15746

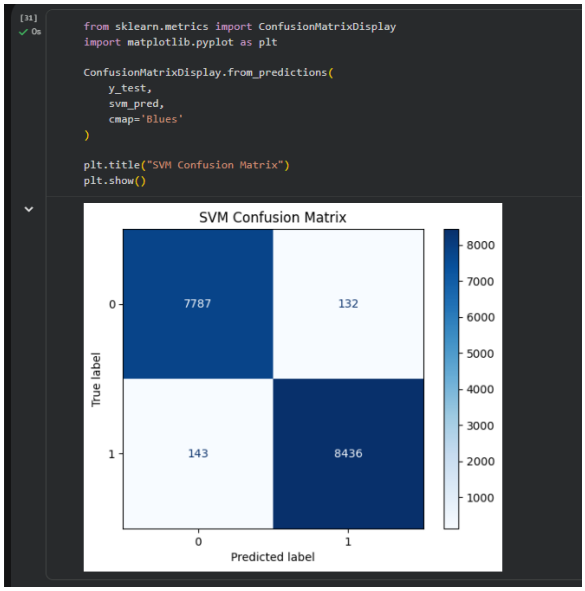

Fig. 3. Confusion matrix Linear SVM.

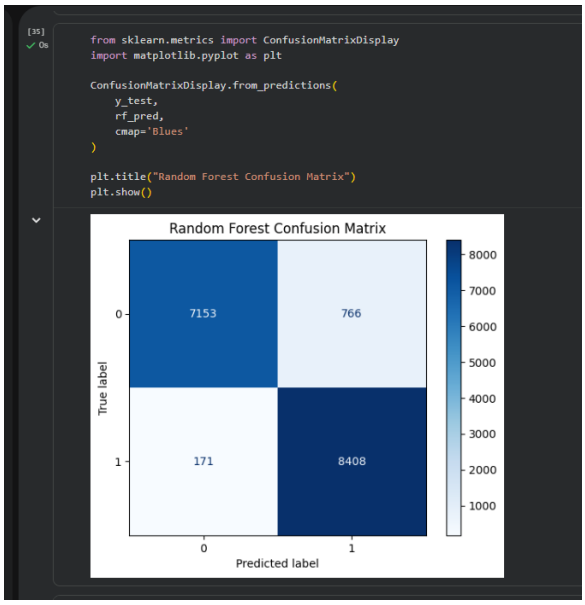

Fig. 4. Confusion matrix Random Forest.

Given its superior performance across all four metrics, the Linear SVM model together with the fitted TF-IDF vectorizer was serialized using joblib for downstream deployment. A lightweight prediction function (predict_email) was implemented to clean, vectorize, and classify arbitrary input text at inference time. When tested on an unseen, realistic phishing message ("Urgent! Your bank account has been suspended. Click here to verify your details."), the deployed model correctly labeled the message as a phishing email, providing qualitative confirmation that the pipeline generalizes beyond the training distribution to the kind of urgency-based social engineering language typical of real-world attacks.

VI. DISCUSSION

The SVM's advantage over Naïve Bayes is consistent with the wider literature reviewed in Section II, where the linear decision boundary of an SVM has repeatedly been shown to model high-dimensional, sparse TF-IDF spaces more effectively than the conditional-independence assumption underlying Naïve Bayes, particularly when discriminative terms (e.g., "verify," "suspended," "click") are correlated with other structural cues in phishing text. The comparatively weaker performance of Random Forest is somewhat more notable: while ensemble tree methods are frequently competitive on structured or lower-dimensional data, they are known to be less well suited to the very high-dimensional, sparse feature spaces produced by TF-IDF vectorization, where individual trees can only split on a small, randomly

DOI: 10.17148/IJARCCE.2026.15746

sampled subset of the 5,000 available features at each node, limiting their ability to exploit the full lexical signal available to a linear model.

Benchmarked against the deep learning and transformer results reported in Section II where CNN-recurrent hybrids and fine-tuned BERT models report accuracies in the 97–99.7% range [4], [5] the 98.33% accuracy achieved by the classical SVM pipeline in this study is highly competitive, and in several published comparisons exceeds baseline LSTM and even some BERT configurations. This supports the position that, for organizations without GPU infrastructure or with strict latency requirements at the mail gateway, a TF-IDF plus SVM pipeline remains a defensible, low-cost choice rather than a strictly inferior fallback to deep learning.

VII. PRACTICAL IMPLICATIONS

For institutions and security teams evaluating phishing-detection tooling, the results suggest a pragmatic decision rule: where computational resources are constrained, a Linear SVM trained on TF-IDF features offers accuracy within one to two percentage points of substantially more expensive deep learning solutions, at a fraction of the training time and with a smaller deployment footprint. The serialized model-plus-vectorizer artifact produced in this study can be integrated directly into a mail-filtering service or browser extension as a first-line classifier, potentially escalating only borderline or low-confidence predictions to a heavier downstream model (e.g., a transformer-based re-ranker) a cascaded architecture that balances cost and accuracy. The precision-recall asymmetry observed for Random Forest also illustrates the importance of selecting an evaluation metric aligned with organizational risk tolerance: a security operations center prioritizing recall (minimizing missed phishing emails) may tolerate Random Forest's higher false-positive rate, whereas a consumer-facing product prioritizing user experience may prefer the SVM's more balanced precision-recall profile.

VIII. LIMITATIONS

Several limitations should be considered when interpreting these results. First, the preprocessing pipeline discards URLs, digits, and punctuation, which removes potentially discriminative signals such as suspicious domain names, IP-based links, and excessive punctuation or capitalization often associated with phishing; a feature set that reincorporates structural and metadata-based cues (sender domain, header anomalies, link count) could plausibly improve on the purely lexical results reported here. Second, the dataset, while large, is a static, historical corpus; phishing language evolves continuously, and periodic retraining or online-learning approaches would be needed to maintain performance against novel campaigns and adversarial paraphrasing. Third, the study did not evaluate robustness to adversarial manipulation (e.g., deliberate misspellings or synonym substitution designed to evade lexical classifiers), an area where transformer-based models have shown measurable advantages in the literature reviewed above. Finally, the models were evaluated only on English-language email text; generalization to multilingual or code-switched phishing content was outside the scope of this study.

IX. CONCLUSION AND FUTURE WORK

This study developed and evaluated a complete TF-IDF-based machine learning pipeline for phishing email detection, comparing Multinomial Naïve Bayes, Linear SVM, and Random Forest classifiers on a merged corpus of 82,486 labeled emails. The Linear SVM emerged as the strongest classical model, achieving 98.33% accuracy and an F1-score of 0.9840, and was successfully packaged into a deployable prediction pipeline validated on an unseen phishing message. These results, read alongside the deep learning and transformer literature surveyed in this paper, indicate that classical, interpretable ML pipelines remain a strong and cost-effective option for phishing detection, particularly where computational resources or latency constraints preclude the use of larger neural architectures. Future work should extend this pipeline in three directions: (i) incorporating structural and metadata features (URLs, sender domain reputation, header anomalies) alongside lexical TF-IDF features; (ii) applying explainable AI (XAI) techniques such as SHAP or LIME to surface the specific terms driving each classification decision, improving analyst trust and supporting regulatory transparency requirements; and (iii) benchmarking the present pipeline directly against fine-tuned transformer models (e.g., DistilBERT) on the same dataset and hardware to quantify the precise accuracy-cost trade-off for this specific corpus, extending the cascaded-architecture concept proposed in Section VII.

REFERENCES

[1] Anti-Phishing Working Group, "Phishing Activity Trends Report, 1st Quarter 2026," APWG, 2026.

[2] A. Alhuzali, A. Alloqmani, M. Aljabri, and F. Alharbi, "In-depth analysis of phishing email detection: Evaluating the performance of machine learning and deep learning models across multiple datasets," *Applied Sciences*, vol. 15, no. 6, p. 3396, 2025.

[3] E. Triana, A. I. Purnamasari, A. Bahtiar, and E. Tohidi, "Improved spam email detection performance based on Naïve Bayes approach TF-IDF vectorizer with multi-metric optimization," *Journal of Artificial Intelligence and Engineering Applications*, 2025.

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 7, July 2026

DOI: 10.17148/IJARCCE.2026.15746

[4] N. Altwaijry, I. Al-Turaiki, R. Alotaibi, and F. Alakeel, "Advancing phishing email detection: A comparative study of deep learning models," *Sensors*, vol. 24, no. 7, p. 2077, 2024.

[5] S. Jamal and H. Wimmer, "An improved transformer-based model for detecting phishing, spam, and ham emails: A large language model approach," *Security and Privacy*, vol. 7, no. 4, p. e402, 2024.