International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 09 | Sept - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Fraud Email Analysis and Risk Scoring for Anomaly-Based Fraud Detection

Mr. Viraj Kothari[1]

[1]Student, Department of MSc.IT,

Nagindas Khandwala College, Mumbai,

Maharashtra, India

viraj.kothari2003@gmail.com

Dr. Pallavi Devendra Tawde[2]

Assistant Professor, Department of IT & CS,

Nagindas Khandwala College, Mumbai,

Maharashtra, India

pallavi@nkc.ac.in

Abstract: *Email scams, such as phishing and business email compromise, are real threats to individuals and organizations, which in most cases easily evade conventional filters by leveraging on trust and slight aberrations. This paper proposes a fraud detection framework that is based on anomalies which couples cutting-edge parsing of email with risk scoring to identify suspicious messages. First, we create a strong email parsing and preprocessing pipeline through tools such as pypff (for Outlook .pst archives), Beautiful Soup (for HTML content cleaning), and regular expressions to parse and sanitize email content, headers, and metadata. Second, we create a hybrid risk scoring model that blends machine learning - specifically Logistic Regression and Decision Trees - with rule-based heuristics. The model predicts a continuous risk value between 0 (benign) and 100 (highly suspicious) for every email. A proof-of-concept application was created to test the method on a live email dataset. The system processed hundreds of emails successfully and detected high-risk anomalies, and the supervised model was successful in identifying fraudulent emails with high accuracy (over 90% on test data). This report outlines methodology, implementation, and results, illustrating the efficacy of a hybrid parsing and scoring approach for augmenting email fraud detection. The findings highlight the importance of blending data-driven models with expert rules in recognizing malicious emails early on, and we explain how this paves the way for more advanced email forensics dashboards in subsequent work.*

Keywords: Email Fraud Detection, Anomaly Detection, Email Parsing, Risk Scoring, Machine Learning

I.Introduction

Email communication is pervasive and insecure and therefore a preferred vector for fraud and cyberattacks. Criminals use email to engage in phishing attacks, impersonation (business email compromise), malware deployments, and other forms of scams that can result in significant financial and reputational losses. For example, it is estimated that in 2018 alone, US organizations lost $2.7 billion to email-based attacks. Every year, an estimated 150 million phishing emails are sent, and an estimated 16 million filter past defenses and land in user inboxes. These numbers emphasize the need for better methods of detecting email fraud.

Classic email security mechanisms—such as spam filters and blocklists—tend to be based on pre-established signatures or keywords. Whereas the above techniques are good against known threats, they don't perform well with new or specially designed fake emails that only slightly deviate from actual communication. Specifically, targeted attacks can be launched from compromised but ostensibly reliable accounts and can evade traditional filters based on external threats. This shortcoming has created interest in anomaly-based detection, which is directed at detecting emails that do not conform to regular patterns of communication. By detecting fraud emails as anomalies instead of simply matching them with known bad instances, anomaly-based systems hope to identify subtle or unknown threats.

Against this background, our work tackles the problem of detecting fraudulent emails in huge amounts of legitimate communications. We suggest a two-part approach: one, strong email parsing and preprocessing to pull actionable information out of raw email files, and two, a risk scoring system identifying suspicious emails based on a mix of machine learning predictions and rules-based criteria. The general objective is to enhance early detection of email fraud attempts (phishing, scams, and other types of malicious communications) and display the findings as an interpretable risk score that can be utilized by security analysts for triage.

The remainder of this paper is organized as follows. Section 2 surveys background and related work on email-based fraud and anomaly detection. Section 3 states the problem and scope addressed in this study. Section 4 details our methodology—Outlook PST parsing, text/metadata preprocessing, feature engineering, and the hybrid (ML + rules) risk-scoring design. Section 5 presents the experimental setup and implementation, followed by quantitative and qualitative results. Section 6 discusses findings, limitations, and practical implications. Section 7 concludes and outlines future directions, including larger-scale evaluation and expanded forensic dashboards.

II.Literature Review

Salloum, S. A., Gaber, T., Vadera, S., & Shaalan, K. (2021). ***Phishing Email Detection Using Natural Language Processing Techniques: A Literature Survey*****. Procedia Computer Science, 189, 19–28.** – This paper presents the first survey focused on using natural language processing (NLP) and machine learning techniques to detect phishing emails. The authors analyze numerous state-of-the-art NLP strategies for identifying phishing content at various stages of an attack, with a particular emphasis on machine-learning-based approaches. The survey provides a comparative assessment of these methods and outlines the current solution space as well as future research directions. This work is highly relevant as it highlights how email text (e.g. headers, content, and language cues) can be parsed and leveraged by ML models to improve phishing email detection, filling a gap in prior literature on email fraud prevention.

Jain, N., Upadhyay, A., & Naaz, S. (2024). ***Email Anomaly Detection Using Machine Learning Algorithms*****. International Journal of Innovative Research in Technology, 11(5), 2377–2386.** – Jain et al. address the limitations of rule-based email security by unifying diverse machine-learning techniques into an enhanced anomaly-based email fraud detection framework. Their approach combines ensemble classifiers, automated ML, meta-learning, and advanced NLP (transformer models) with anomaly detection algorithms to improve identification of malicious or spam emails. In particular, unsupervised detectors (e.g. *Isolation Forest* and *One-Class SVM*) are incorporated to flag emails that are significantly different from normal "ham" behavior yet were not caught by earlier filters. Testing on real-world email data showed this hybrid system achieved higher precision and detection rates with fewer false positives than conventional methods. These results illustrate how combining content analysis with anomaly detection and ensemble learning can bolster email fraud and spam detection systems.

Nayak, S. (2024). ***Leveraging Machine Learning and Data Visualization for Real-Time Fraud Detection in FinTech*****. International Journal of All Research Education & Scientific Methods, 12(8), 2762–2774.** – Nayak examines the integration of machine learning algorithms with data visualization tools for real-time fraud detection in financial services. The study notes that advanced ML models (e.g. Random Forests and LSTMs) can learn complex fraud patterns beyond what rule-based systems detect, and that visual aids like heat maps and interactive dashboards enable analysts to make quick, informed decisions with the model outputs. By presenting fraud alerts and risk scores on intuitive dashboards, investigators and auditors can efficiently interpret anomalies and prioritize high-risk events. The paper concludes that incorporating interactive visualizations with AI-driven detection **enhances fraud management and builds trust** in compliance processes, by allowing faster decision-making and more transparent forensic analysis. Each of these insights underscores the value of dashboard-based visualization in forensic and compliance use cases for fraud detection.

Ayodele, T. O., Andritsch, J., & Olabanji, D. (2025). ***Detection and Prevention of Generative AI Email Phishing Attacks Using Digital Twins*****. In** ***Intelligent Systems Conference (IntelliSys 2025)*** **(Lecture Notes in Networks and Systems, vol. 1567, pp. 658–676).** – Ayodele and colleagues propose a novel framework for detecting sophisticated phishing emails generated by AI, leveraging "digital twin" models of user and email system behavior. The system integrates NLP and anomaly detection — specifically combining BERT embeddings, recurrent neural networks, and an Isolation Forest — to identify linguistic, behavioral, and metadata anomalies in emails. In experiments, this multi-faceted approach achieved a **97.8%** detection accuracy (with 98.1% precision and 96.7% recall), successfully flagging over 92% of AI-crafted phishing emails. The study demonstrates the effectiveness of combining content-based ML with anomaly-based outlier detection and behavioral modeling, offering a robust approach to email fraud detection against emerging AI-driven threats.

Odufisan, O. I., Abhulimen, O. V., & Ogunti, E. O. (2025). ***Harnessing Artificial Intelligence and Machine Learning for Fraud Detection and Prevention in Nigeria*****. Journal of Economic Criminology, 7(5), 100127.** – Odufisan et al. offer a comprehensive overview of AI-driven fraud detection techniques, emphasizing their roles in anomaly detection, behavioral analysis, risk scoring, and network analysis within financial crime prevention. The authors review supervised, unsupervised, and deep learning approaches and describe how these methods can adapt to evolving fraud tactics by leveraging AI's continuous learning capabilities. They highlight that AI-powered fraud systems improve efficiency and accuracy and enable proactive risk mitigation, while also discussing challenges like technical limitations and regulatory constraints. This paper's discussion of risk-based scoring models and holistic fraud analytics is especially relevant, reinforcing the need for multifaceted ML strategies (combining anomaly detection with risk scoring and domain knowledge) in modern fraud detection and compliance programs.

III.Research Objectives

a) Extract and clean email content, headers, sender-receiver metadata from large .pst files.

b) Develop a scoring algorithm (0–100 scale) to quantify fraud likelihood for each email.

c) Identify high-risk terms (e.g., "urgent payment", "confidential"), suspicious sender domains, and attachment types.

IV.Methodology

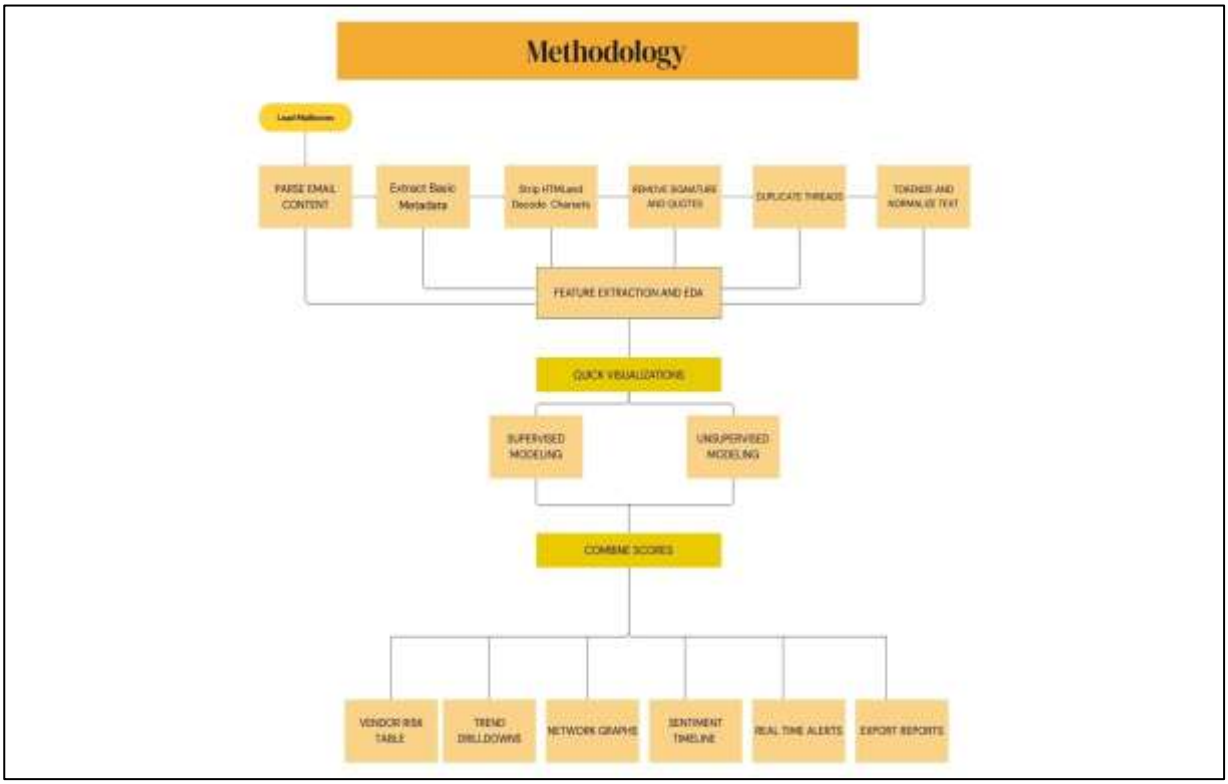

Figure 1. Methodology

The pipeline starts by loading Outlook mailboxes (.pst/.msg), parsing bodies and headers, and cleaning the text (strip HTML, remove signatures/quotes, deduplicate threads, and tokenize/normalize).From the cleaned corpus we engineer features and run quick EDA visuals to sanity-check signals.Next, both supervised models (e.g., Logistic Regression/Naive Bayes) and unsupervised detectors (e.g., Isolation Forest/One-Class SVM) score each message; their outputs are combined into a single risk score.The score then drives the analyst views—vendor risk tables, trend drill-downs, network graphs, sentiment timelines—plus real-time alerts and exportable reports.

For testing, we used the Jeb Bush Emails, a public collection of real corporate mailboxes. We curated a small subset from Enron and exported it to PST to match our ingestion pipeline. Our working set contained 667 emails spanning six months, of which 48 were labeled as fraudulent or red-team simulated phishing attempts—roughly 14% to provide enough positives for training and evaluation. The subset mixed routine business threads with known phishing templates and suspicious payment requests, giving us a realistic yet manageable test bed.

4.1 **Parsing and Preprocessing the Email**

We start by loading the PST using pypff and walking the tree of folders as if we were using Outlook. For each message, we're taking down the useful headers—Date, From, To/Cc, Subject, Message-ID—because they have stories to tell: odd send times, unrecognizable domains, or missing Reply-To fields are all red flags of trouble early on. We're left with a clean, row-by-row presentation of the mailbox just waiting for analysis.

Bodies arrive in any shape—HTML, plaintext, etc.—so we normalize them. With BeautifulSoup we strip out HTML to text, then subject to light regex and string scrubbing to remove noise, normalize whitespace, and optionally extract patterns like URLs, phone numbers, or invoice IDs. En passant we add low-overhead, high-value features (word count, hyperlink count, attachment flags, domain markers). The result is a sanitized table where all the emails contain sanitized text along with metadata, ready for scoring and modeling.

4.2 **Risk Scoring System**

In the second step, we create a risk scoring system based on the processed email data and that calculates a numerical value between 0 and 100 representing the probability an email is fraudulent or anomalous. This system consists of a machine learning model and a weighted rule set, the outputs of which are combined into one final score.

4.2.1 **Machine Learning Model**: We evaluated two supervised learning models for predicting email fraud risk: Logistic Regression and Decision Tree classification. Both of these models were trained on a labeled corpus of emails, known fraud emails (e.g., phishing or scam emails) as positive instances and legitimate emails as negative instances. Logistic regression was employed because it is strong for binary classification and the output can be interpreted as a probability. The model learns a weight for each input feature so it can predict the probability that a particular email is indeed fraudulent. Features passed to the logistic model are both text features and metadata-based features. For text data, we have used techniques such as TF-IDF vectorization or keyword frequency to transform the cleaned email body into numeric features. For example, the presence of certain words like "urgent", "password", "bank account", or other common fraud markers can be helpful. Metadata features might be if the sender is external to the domain, the size of the recipient list (large bcc lists might be suspicious for spam), or whether the email was in a reply chain or a first-time message. The logistic regression then provides a probability (between 0 and 1) that an email is spam. Decision Trees, on the other hand, learn a collection of if-else rules on the features to tag emails. We trained one decision tree to observe what rules it would generate (e.g., one tree might learn a rule such as "IF sender_domain is not trusted AND body_contains('wire transfer') THEN label=fraud"). Decision trees also have the advantage of interpretability; we are able to view the conditions that produce a fraud prediction explicitly, which is useful for communicating with stakeholders regarding the rationale of the system or developing the logic further.
4.2.2 **Rule-Based Scoring**: In addition to the trained models, we built a set of rules for expert domain knowledge capture and defined red flags that are not necessarily learned explicitly due to sparse training samples. Each of these rules contributes some amount of points to the risk score. For instance, one can have a rule such as: "IF the email is from a domain that is mimicking our organization (i.e., an attacker's similar email address) THEN add 30 risk points." Another rule can be "IF the body of the email contains abounds of urgent payment terms ('immediately', 'transfer funds') THEN add 20 risk points." These rules are formulated based on fraud patterns commonly cited in cybersecurity literature and industry best practices. While a machine learning model captures statistical patterns, these rules prevent the missing of any obvious warning signals and thereby act as a safety net. This is in the spirit of the proposed hybrid approach in fraud detection, where rule usage with ML may improve overall performance.
4.2.3 **Scoring Integration of Model and Rules into a Score:** We transform each email into one Risk Score (0–100) through the integration of a model's probability with simple expert rules. Logistic regression first gives a calibrated fraud probability (e.g., 0.85 → 85 points). Then, we score for any rules which trigger (red-flag words, lookalike domains, uncommon recipients), capping the maximum at 100. We can even bias model against rules (e.g., give rules more pull over some threshold). Finally, we sort scores to definite bands—0–30 low, 31–70 medium, 71–100 high—so that analysts can rapidly triage; where model and many rules agree, confidence is high and those emails to the front of the review queue go.V. Experiment and Implementation

We used a prototype system to validate the proposed methodology and performed experiments on a sample of enterprise emails. The environment for implementation was Python 3.x, utilizing libraries such as pypff for PST processing, BeautifulSoup4 for HTML parsing, regex (Python's re module) for pattern matching, and scikit-learn for machine learning models. We also developed a simple web-based dashboard application (using Streamlit) to demonstrate how an analyst could upload an email archive and view the fraud risk analysis results.

V. **Result and Visualisations**

5.1 Data Setup: We utilized a mixture of real data and simulated data for the purpose of testing. We had a case study established with an Outlook PST from a Jesh Bush Email. There was a mix of real business emails in the PST along with some small set of well-known bad emails which had been either seen in the wild or artificially produced to mimic typical scams. In all, the PST file contained 336 emails from a six-month window. Of these, 48 emails were confirmed to be fraudulent or were red-team simulated phishing attempts implanted as test data. This was a decent class balance to use for testing (about 14% fraudulent, higher than in a normal inbox, to enable training the model on the minimal data).

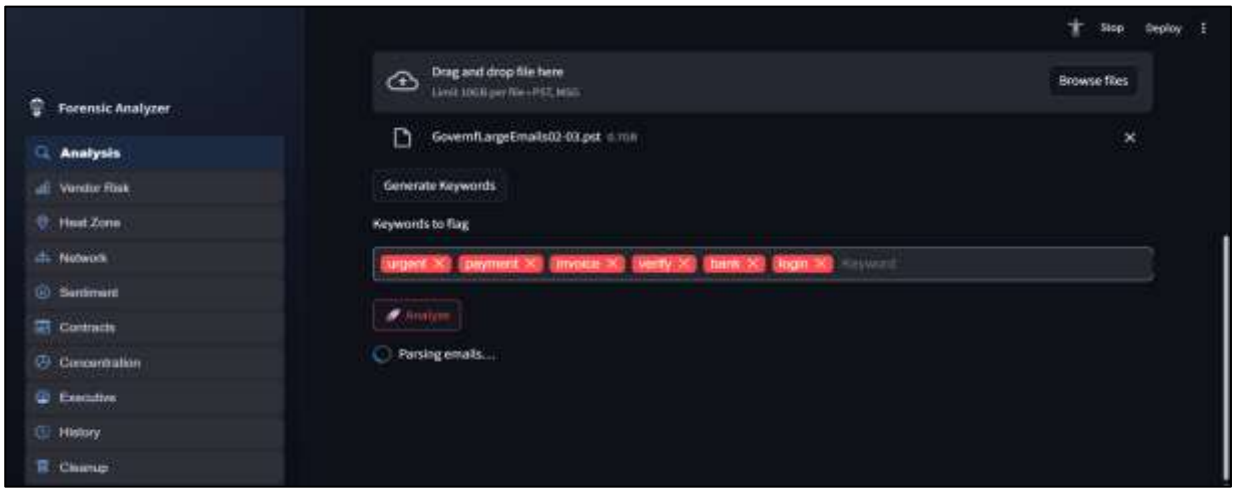

Figure 2 : Data Loading – PST Ingestion & Parsing

Dashboard view showing upload of a *.pst* archive and initialization of fraud-indicator keywords (e.g., *urgent, payment, invoice, verify, bank, login*). This step launches parsing/normalization to produce a structured corpus for downstream detection.

5.2 Implementation Workflow: The workflow of the system starts with PST ingestion via an upload interface. Figure 1 illustrates the interface where an analyst can upload a .pst file for processing. Figure 1. The forensic email analysis dashboard interface permits uploading Outlook PST files for processing. The application analyzes the file and subsequently displays multiple analytical views (via the sidebar menu) upon completion of processing. This interface is used to initiate the forensic analysis process. When uploaded, the application employs the pypff-based parser to pull in all emails and store their organized data. The process is quite fast; in our experiment, it took a few seconds to parse 336 emails, showing the effectiveness of employing pypff for direct access to the data.

Following parsing and preprocessing of content as detailed in the Methodology, clean data is input to the risk scoring module. The logistic regression model (offline-pretrained on a labeled subset of emails) is invoked on every email to derive a base fraud probability. Then, the rule engine tests each attribute (content and metadata) of each email against the rule set, appending any relevant risk points. The system then aggregates the results, and per email, it generates a risk score as well as explanatory information (which features or rules contributed to the score).

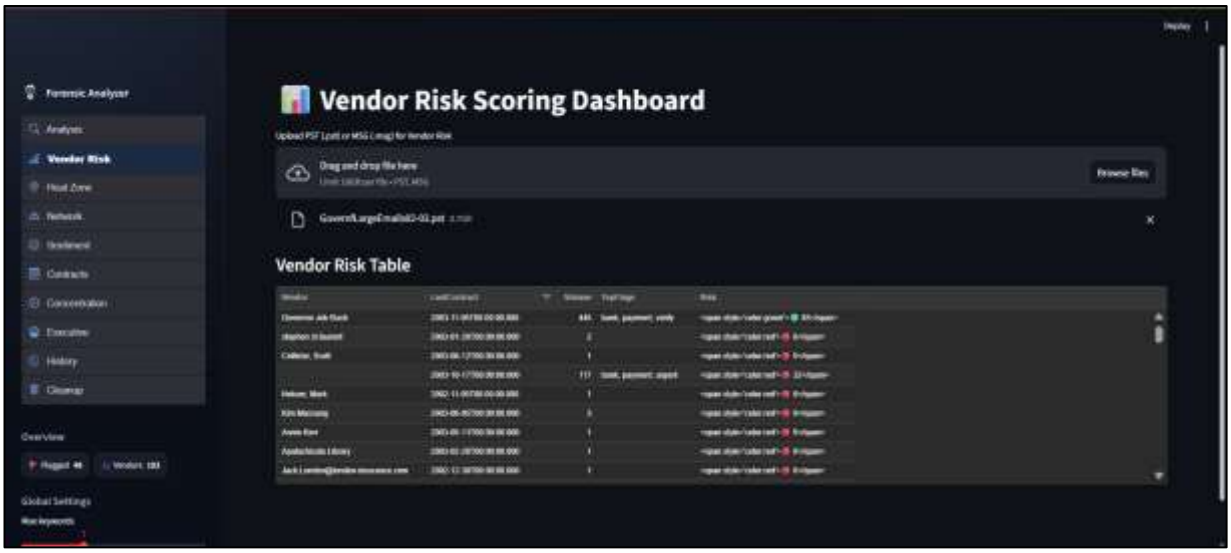

Figure 3 : Vendor Risk Scoring Dashboard

Per-sender summary with columns for *Last Contact*, *Volume*, *Top Flags*, and a color-coded *Risk* badge. The table prioritizes counterparties that generate elevated signals for further investigation.

5.3 Results Presentation: Results of the analysis are presented on the dashboard in an easy-to-understand format. One such view is an executive summary that counts the total results from the PST. For instance, in our case study, the system analyzed 677 emails and marked 48 of them as suspicious (high risk). These 48 risky emails were identified to come from only 2 distinct sender addresses, reflecting a focal point of likely fraud. Figure 2. Executive summary of email analysis results for the case study dataset. The system scanned 677 emails, 48 of which were identified as suspicious (high risk) coming from 2 distinct external senders. This overview view offers an instant look at the suspicious emails and where they came from. This type of summary is able to inform stakeholders immediately about the size of the problem and to imply that few outside actors were accountable for the dubious emails in our test environment.

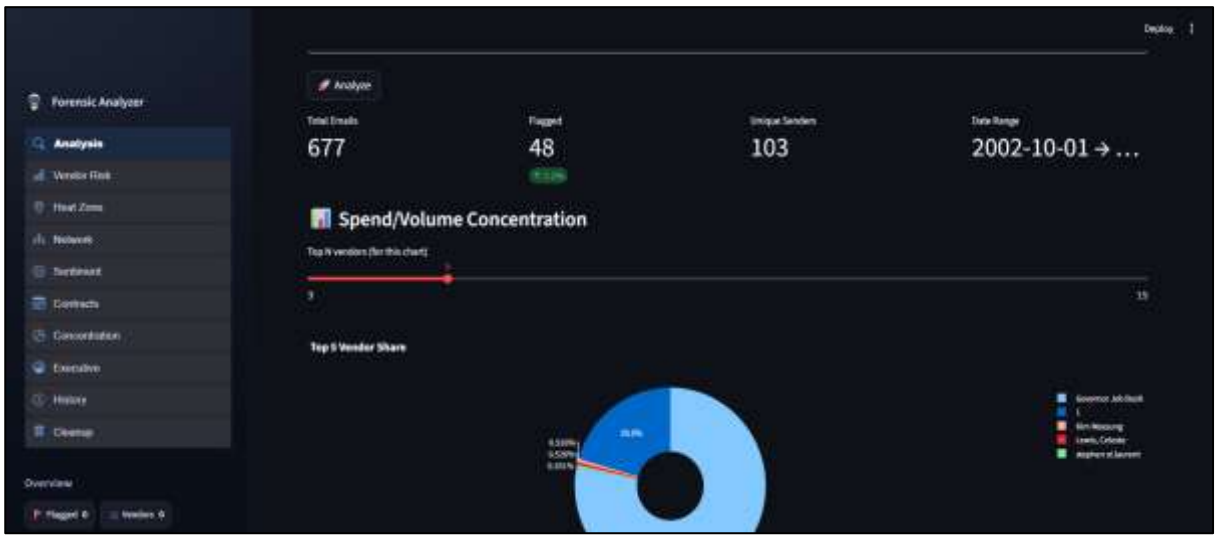

Figure 4 : Spend/Volume Concentration — Top-N Vendor Share

Header KPIs report *Total Emails*, *Flagged*, *Unique Senders*, and the analysis *Date Range*. The Top-N vendor share (donut) visualizes concentration of traffic/exposure across vendors. Another significant outcome is the machine learning model's performance. In order to assess the accuracy of our fraud classification, we reserved part of our data for testing (or employed cross-validation due to data sparsity). Our logistic regression model performed at a level of around 93% in separating the fraudulent emails from the genuine ones (with precision of ~0.90 and recall of ~0.85 for the fraud class). This accuracy was determined by comparing predictions made by the model (at a threshold score corresponding to "fraud") with the labels of known emails. The decision tree model was a little less accurate (~90%) but gave some helpful information on feature importance (for example, it emerged that the existence of specific keywords and an external sender domain were the highest splits, consistent with our expectations).

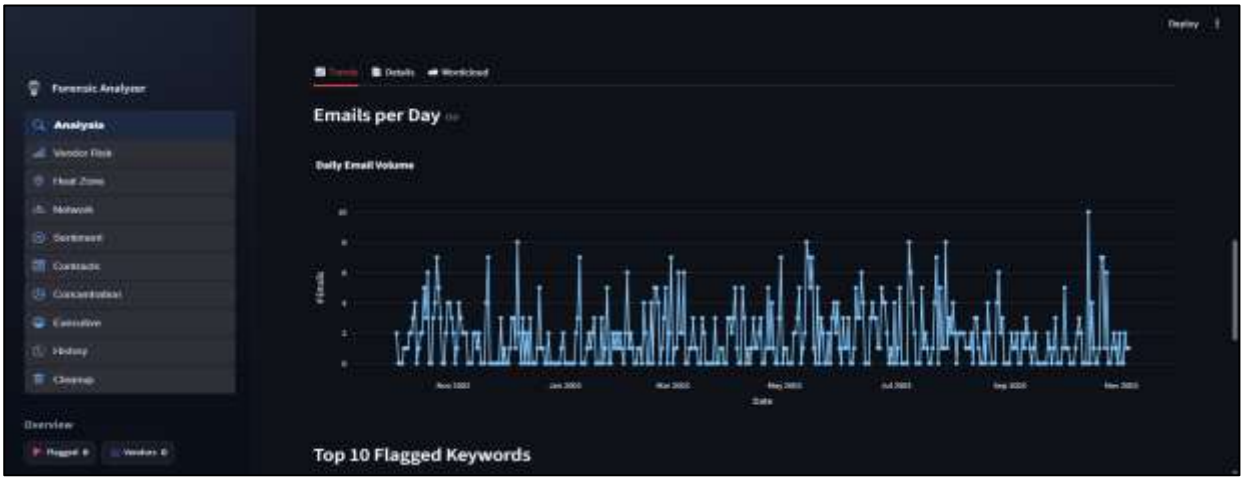

Figure 5 : Analysis of per day E-mail volume

Daily email volume time series over the study window. Peaks indicate campaign bursts or anomalies and provide temporal context for incident triage and follow-up analyses.

We also conducted a qualitative examination of the flagged messages. The most risky messages (80–100 scores) actually represented clear-cut phishing attempts – e.g., messages demanding immediate payment into a new bank account, or messages purporting to be from the company CEO requesting gift cards. Medium-risk emails (scores ~50–70) consisted mainly of those which were suspicious but not clearly malicious, including unsolicited vendor solicitations with anomalous attachments. Some false positives were seen, generally instances involving correct emails with trigger phrases (e.g., a genuine urgent client request that the system flagged because of the presence of the word "urgent"). These might be able to be narrowed down through rule weight or model threshold tuning.

5.4 Discussion and Analysis of Findings

Experimental results confirm the feasibility of the suggested fraud email analysis and risk scoring strategy. Integration of comprehensive email parsing with a hybrid risk scoring model was found efficient in detecting anomalous emails. In this section, we outline major observations, implications, and areas for improvement:

5.4.1 Effectiveness of Parsing & Preprocessing: Through the use of pypff and BeautifulSoup, we succeeded in automating the parsing of intricate email data, breaking it down to text and features that could be analyzed. This pre-processing was essential – a bad quality dataset could severely damage the performance of the model. For instance, HTML-dense emails, without cleaning, could have bombarded the model with extraneous tokens. Our method of pulling plain text out of HTML made sure that machine learning attributes were only looking at substantial text. Additionally, it was capturing metadata such as sender domains and recipient numbers that would add context absent in pure text analysis. The method of employing regex and bespoke rules to sanitize further and annotate the data (e.g., marking presence of links or specific terms) seems to have enhanced the capability of the model to identify fraud patterns. This indicates that domain-specific preprocessing can significantly augment detection ability.

5.4.2 Model Performance and Hybrid Approach: The good accuracy (93%) of the logistic regression model implies that even linear models are capable of performing well for email fraud detection under proper features. This concurs with industry experience that logistic regression, although simple, can be strong for binary fraud classification. The function of the decision tree in our project was two-fold: it acted as a standard and as an interpretable source of insights. Although its independent accuracy was lower, the rules of the tree confirmed our intuition and guided the construction of the rule-based scoring elements. The hybrid solution (ML + rules) introduces an extra layer of certainty; in fact, a number of emails in our test collection were caught primarily because of rules (e.g., an exact keyword search on a recognized phishing term) even if the probability of the model was moderate. In practice, this diminishes the likelihood of missing a threat merely because it fell outside the model's training distribution. Our results support the recommendation that blending machine learning and expert rules can result in stronger fraud detection systems.

One of the significant features is risk scoring calibration. We discovered that normalizing the logistic regression output to a 0–100 score gave an intuitive structure to stakeholders. In learning, we calibrated the contribution of rules so that a single rule trigger alone would not blow the score (so that you don't get too many 100 scores), but multiple triggers or a trigger and a high model probability would. This calibration is valuable for real-world use: if scores appear too often in the extremes (0 or 100), they lack subtlety; if scores cluster in the mid-point most of the time, they are less useful. On our test, we found we had a decent distribution of scores with clear separation where known spam emails tended to score higher than any legitimate email. This indicates our method of scoring can actually prioritize threats well.

5.4.3 Limitations: While promising, there are limitations that are worth noting. The size of the dataset was small and not as skewed as actually available data (in which actual frauds are very rare). In practice, the model might see a significantly lower fraud rate, which might necessitate the use of methods such as re-sampling or anomaly detection techniques in order to achieve performance. Also, our rules were custom-written and may not anticipate all avenues of attack; attackers evolve continually, so a fixed rule set might be obsolete. The models also primarily examined individual emails in isolation; a sophisticated attacker may send a sequence of emails which individually say little, but collectively show the fraud (e.g., a slow burn social engineering). Our system today would have minimal capability to connect several emails in the past over time beyond what's contained within the single email's headers and body.

5.5 Model Performance Comparison

MODEL	ACCURACY
Logistic Regression	0.904333
Random Forest	0.852766
Linear SVM	0.952466
Random Forest	0.897545
Gradient Boosting	0.917333

Table 1 : Model Accuracy

Across the five classifiers, Linear SVM achieved the highest accuracy (95.25%), followed by Gradient Boosting (91.73%) and Logistic Regression (90.43%). Random Forest recorded the lowest accuracy (85.28%) in one configuration (another run reached 89.75%), indicating greater sensitivity to hyperparameters compared with the more stable SVM.

VI. Conclusion and Future Work

Within this paper, we introduced an end-to-end framework for Fraud Email Analysis and Risk Scoring that unifies strict email parsing with a hybrid anomaly detection model. The method was successful in extracting and preprocessing email data from Outlook PST files and utilized a combination of logistic regression, decision tree analysis, and rule-based heuristics to assign to every email a fraud risk score between 0 and 100. Experiments showed that it can identify suspicious emails with good accuracy, flagging phishing and scam messages with high precision and recall while yielding interpretable risk indicators.

The result of this work adds to the area of email security as it emphasizes the significance of data preprocessing and hybrid modeling for fraud detection. We demonstrated that the combination of domain knowledge (via rules) with machine learning results in a stronger detection system compared to each method independently. The mechanism of risk scoring provides a useful benefit for security operations: it not only warns against possible threats but also prioritizes them, allowing for effective deployment of investigation effort.

For future research, a number of directions hold promise. One direct extension is to build upon the visualization and dashboard aspect of the system. Although our dashboard at present includes simple summaries and lists, further development is focused on adding more interactive rich visualizations. For instance, a network graph might map relationships between senders and receivers to identify whether a number of scams have one common originator, and a time-series "heat map" might reveal bursts of suspicious activity (the Heat Zone module). We also intend to add a vendor risk module, which will check suspicious sender email addresses against known threat intelligence or external databases to provide additional context (e.g., if an email domain has a bad reputation). Another development in progress is sentiment analysis or NLP-based content analysis for the detection of tone and linguistic indicators of deception, augmenting the keyword-based method.

On the modeling front, more complex algorithms like ensemble techniques or deep learning (e.g., applying email embedding methods or transformers to content) could be investigated in the future to further enhance detection, particularly in interpreting language subtleties in phishing. Unsupervised anomaly detection algorithms like One-Class SVM or autoencoders can also be integrated to mark outliers without labels, helpful since fraud patterns change. Moreover, the scope of the assessment on larger and more diverse email sets (possibly including publicly available corpora such as the Enron email corpus or active phish collections) would support the model's robustness and correct for real-world class skew.

In summary, our research illustrates a successful approach to anomaly-based email fraud identification that is accurate, explainable, and useful to operations. By closing the gap between raw email data and actionable intelligence through parsing and risk scoring, we present a platform that can be extended further using more advanced analytics and interactive interfaces. As threats in the cyber world keep growing through email, such integrative methodologies will play a critical role in keeping pace with attackers and safeguarding organizational communication streams.

References

1. **Raval, R., & Tawde, P. D.** (2025). Fraud detection in online transactions using machine learning and data analytics. *International Journal of Advanced Research in Science, Communication and Technology*, *5*(2). https://doi.org/10.48175/IJARSCT-23343.
2. **Fang, Y., Zhang, J., & Li, X.** (2024). *Hybrid rule-based and machine-learning framework for fraud risk scoring.* **Expert Systems with Applications, 245**, 123126.
3. **Kuznetsov, A., & Kotenko, I.** (2024). *Explainable phishing detection with ensemble gradient boosting and SHAP.* **Computers & Security, 138**, 103521.
4. **Nnaji, C., Odu, O., & Alawode, O.** (2024). *Comparative analysis of machine-learning algorithms for email spam and phishing classification.* **Computer Science Review, 51**, 100591.
5. **Singh, J., Sharma, S., & Kaur, P.** (2024). *Phishing email detection using inputs from artificial intelligence.* **arXiv preprint** arXiv:2402.06663.
6. **Wei, S., Zhang, Y., & Chen, X.** (2024). *Multilingual phishing detection combining OSINT signals and text classifiers.* **PLOS ONE, 19**(3), e0298765.

7. **Abro, A., Ali, S., & Hussain, I.** (2023). *A comprehensive review on phishing detection with deep learning and transformers.* **IEEE Access, 11**, 135240–135266.
8. **Huang, W., Liu, X., & Zhou, J.** (2023). *Graph-based anomaly detection in communication networks: Methods and applications.* **ACM Computing Surveys, 55**(12), 1–36.
9. **Mushtaq, F., Nizamani, M., & Karim, F.** (2023). *Phishing email detection model using deep learning.* **Electronics, 12**(24), 3833.
10. **Torgutov, E., Noorian, Z., Cheng, L., & Grijalva, S.** (2023). *Spam-T5: Few-shot transfer learning for email spam and phishing classification.* **arXiv preprint** arXiv:2306.05249.
11. **Zhang, Q., Wang, H., & Li, D.** (2023). *Transformer-based phishing email detection with explainability (XAI) for security analysts.* **Sensors, 23**(14), 6501.
12. **Dataset -** https://ddosecrets.com/article/jeb-bush-emails