InternGuard: Fake Internship Offer Detection System

Junaid Ahmed Khan[1], Goudicherla Chinmayi Srija[2], Dr. B. Poornima[3], Dr. K. Sreekala[4]
[1,2]*Undergraduate Students, Mahatma Gandhi Institute of Technology*
[3,4]*Assistant Professor, Mahatma Gandhi Institute of Technology, Hyderabad, Telangana, India*
doi.org/10.64643/IJIRTV12I12-203203-459

Abstract— **Fraudulent internship offers emails targeting undergraduate students have increased significantly with the growth of online recruitment platforms. These emails often impersonate legitimate organizations and attempt to deceive students into paying registration fees, security deposits, or onboarding charges. This paper presents InternGuard, an intelligent multi-module system designed to detect fraudulent internship emails through a combination of email header forensics, machine learning-based content analysis, and website legitimacy verification. The proposed system evaluates SPF, DKIM, and DMARC authentication records to identify spoofed emails, while TF-IDF feature extraction and a Random Forest classifier are used to analyze email content. In addition, domain age and SSL certificate validation are performed to assess the credibility of embedded websites. Experimental evaluation showed that the Random Forest model achieved the best overall performance among the tested classifiers, obtaining an F1-score of 86.3% and a cross-validation accuracy of 85.26%. External validation on the EMSCAD dataset demonstrated reasonable generalization capability despite domain differences. The system was further evaluated using real-world fake internship emails collected from students, where it successfully identified multiple scam indicators and classified both emails as high risk. InternGuard also incorporates explainable risk scoring and rule-based scam phrase analysis to improve transparency and reduce false positives. The complete system is implemented as a Flask-based web application with an interactive user interface designed for practical student use.**

Index Terms— **Fake Internship Detection, Phishing Email Analysis, Random Forest, TF IDF**

I. INTRODUCTION

Fraudulent internship recruitment emails are being sent to undergraduate students via internet internship platforms and social media sites. Generally, these scams act as representatives of well-known organizations and try to coerce students into providing registration fees, refundable security deposits, onboarding fees or payments to verify documents. Recent cybersecurity reports show that many students in India have come across suspicious internship offers, with some victims losing money after giving a reply to such emails.

These scams are even more difficult to detect because many fraudulent emails are very well-crafted to look like a legitimate HR message. Scam messages may feature personalized greetings, professional language, corporate branding and/or urgent deadlines meant to pressure the recipient to take immediate action. Attackers also often use free email servers like Gmail or Yahoo to send emails through and seem legitimate to simple spam filters.

Spam and phishing detection methods, such as traditional spam filters, and website scanner products based on blacklists, are not designed to detect internship recruitment fraud. The majority of the available systems are general spam catcher systems which are inadequate to detect the domain specific spam which is usually seen in the internship scams happening in India with student interns. To overcome this drawback, the present paper introduces InternGuard, a multi-layered detection framework that integrates email header forensics, machine learning-based content analysis and website legitimacy verification all within one decision engine.

We have made the following contributions: (i) we have created a domain dataset of 10,000 labelled internship emails; (ii) we have designed a negation-aware scam phrase detector that contains more than 60K scam phrases and a few rules to detect the presence of red flags.

II. LITERATURE SURVEY

Studies on detecting fraudulent online messages have seen numerous methods, such as spam detection and detecting phishing emails, fraudulent job ads, and the

credibility on a domain. We here provide and will comment on pertinent prior studies in each of these threads.

Machine Learning in Spam and Phishing Detection: Bhowmick and Hazra performed experiments with five machine learning classifiers comparing their performance on multiple spam data sets, and found that the overall performance of the Random Forest was the best, with an average F1-score of 0.96 on sparse TF-IDF-based feature vectors. This is mostly because it is in an ensemble form which aids in reducing variance whilst maintaining low bias - an essential element in working with large sets of features in text. In a similar fashion, Salloum et al. [2] employed a Linear SVM model with TF-IDF features to identify phishing emails, and augmented it with a mechanism (negation-sensitive) phrase matching. Their approach had also impressive results with a 96.2% accuracy in SpamAssassin Public Corpus.

Employment Scam Detection: Ilias et al. created the Employment Scam Aegean Dataset (EMSCAD), a dataset of 17,880 valid and scam listings. They used this dataset to train a baseline Logistic Regression model with a 89 percent accuracy with characteristic features of scams such as promises of unrealistic rewards, vague job description, and lacking corporate information or logos. Building on these insights, Alghamdi and Khan [4] used a Random Forest model with TF-IDF features from the same dataset to further increase accuracy to 97.8%. Their work also attests the appropriateness of tree-based ensemble models in the battle against employment scams.

Email Authentication Mechanics: Kucherawy and Zwicky present the details of the three most commonly used email authentication mechanisms - SPF, DKIM and DMARC - and their importance in email security. Their study revealed that those companies that employed all three of the mechanisms could decrease the level of spoofing by over 80 which is quite encouraging. Building upon this study, Nakamura also researched inconsistencies in email headers and found out that hackers usually manipulate front-end field like the From field but would neglect back-end fields such as the Return-Path field, the Authentication-Results field, and so on. Such inconsistency offers unique patterns which could be utilized to identify spoofed emails.

Domain and Website Analysis: Mohammad et al. analysed 11,055 phishing and benign URLs, and found that almost 91% of phishing domains were only registered within a year of use. By taking into consideration domain age and SSL authenticity, they were able to achieve a classification accuracy of 94.2%. This study presented some useful information in the analysis of the webpage of the proposed solution.

Interpretability of the Context of AI: Guo et al.sub-optimized DistilBERT on phishing email datasets and achieved an impressive accuracy of 98.4 per cent, outperforming the accuracy of the conventional TF-IDF-based methods. Attaining higher accuracy is accompanied with higher computational burdens of fine-tuning. Equally, as pointed out by Ribeiro et al., model explainability is necessary. Using LIME and SHAP, they demonstrated that fraud detectors that provide word-level explanations boost user confidence and promote adoption. This study gave the idea to integrate explainability in InternGuard.

III. DESIGN METHODOLOGY

The design of Intern Guard is based on parallel analysis with three modules and a weighted decision engine that provides final classification. The system accepts 2 forms of input: the user can simply paste the raw text of an email into the web interface or the user can upload the contents of a .eml file into the web interface to allow detailed analysis of the email and its associated headers. This loosely defined input design makes it possible to make sure that the system is able to respond to simple and sophisticated use cases effectively. Figure 1 shows the overall workflow and communication between these elements.

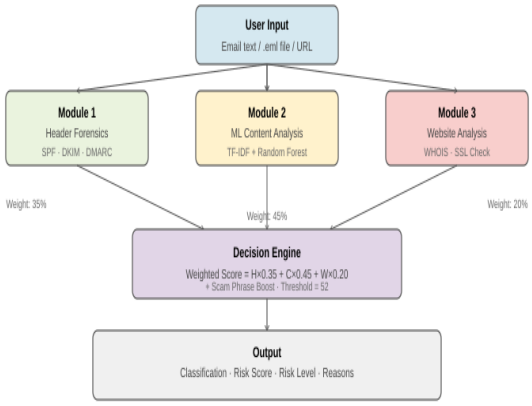

Figure I: System Architecture of InternGuard

A. Dataset Construction

To cover a wide range of general English email patterns as well as particular internship scam patterns, the training data set was gathered from four different sources. In the first place, the samples were obtained through the SpamAssassin Public Corpus, including spam and non-spam ("ham") emails based on raw .eml emails using Python email library. Second, we utilized the Kaggle Email Classification dataset, which has emails in the word-frequency form, reformat them into text. Besides these public datasets, the data of the domain were included in order to be relevant. This dataset included 1,000 templates of hand-written emails and modeling the typical tricks used in the Indian market to defraud the company with its security deposit, UPI transactions, registration fees, and last-minute changes to the cancellation policies. Finally, to depict the actual scam cases, 500 real fake internship emails sent to students were included.After removing duplicates, minimum length filtering (30 chars) and shuffling, our final dataset consisted of three to four labeled emails per class with a similar number of fake and genuine emails. This ratio not only makes a model reliable, but also it does not overweight a class.

B. Text Preprocessing Pipeline

All email samples were subjected to a standardised email preprocessing pipeline to enhance the quality of the feature extraction and to minimise textual noise. First of all, various ways of writing a monetary value, like “₹500”, “Rs.500” and “INR 500” were converted into a uniform format. To guarantee uniformity in the token analysis, all the text was then made lower case.

The sentences were further preprocessed by removing punctuation, tokenizing the sentences, removing stopwords and lemmatizing them using the NLTK library. Other non-informative special characters, excessive spacing and URLs were also stripped. These operations were used to reduce the dimensionality of the data and retain meaningful patterns of the email texts related to the scam.

A specific preprocessing pipeline was developed to preserve phrases that are commonly used in internship scams, like phrases concerning payment requests, urgency, and internship procedures. This further enhanced the performance of the following TF-IDF feature extraction and machine learning classification.

C. Feature Extraction — TF-IDF

The text of the email, after cleaning, is converted to a numerical one using the TF-IDF (Term Frequency - Inverse Document Frequency) approach. This approach is more focused on the frequently occurring words in a specific email but not in the larger email corpus, and is very applicable to such a corpus as spam detection. The limit of the word vectorizer is 10,000 features, unigrams, bigrams and trigrams. The latter is particularly important, to make sure that the next phrases are recognized as potential scam phrases: failure to comply will.

D. Module 1 - Header Forensics

This module examines the authenticity of e-mails with the help of email header. The analysis system is based on Python built-in module email, which is used to retrieve the information in the header of an email file (in .eml format). Checks authentication data of SPF, DKIM and DMARC using extraction with regular expressions. These tests may be used to determine if an e-mail is a spoof or suspect email, and the risk scores are calculated by rules using the various tests as listed in Table I.

Table I: Header Analysis Scoring Rules

Check	Condition	Risk Added
SPF	Result = fail	+30 pts
DKIM	Result = fail	+30 pts
DMARC	Result = fail	+20 pts
Domain mismatch	From ≠ Return-Path	+20 pts
Display spoof	Name = company, domain ≠	+25 pts
Max header score	—	100 pts

E. Module 2 — ML Content Analysis

In this study, the machine learning pipeline in Figure I was tested. We tried five algorithms with 5-fold stratified cross-validation to determine the most suitable model to use in this purpose. Among them, the Random Forest classifier was identified to do the best and hence one was used in the final model.

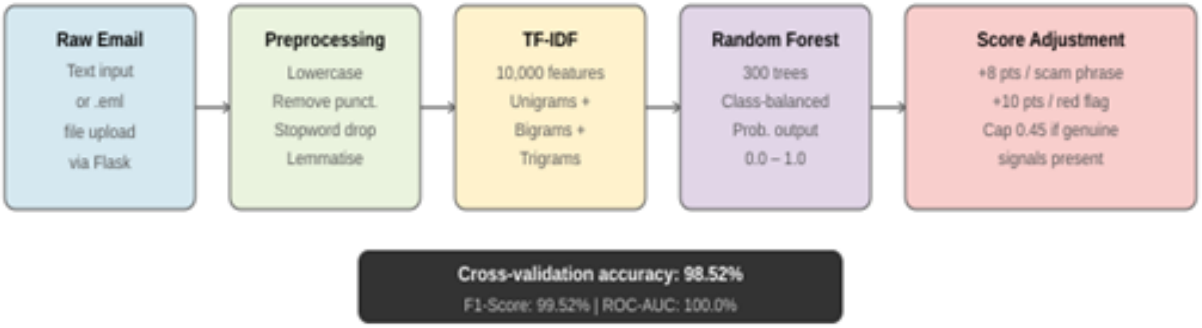

Figure I: Module 2 - ML Content Analysis

To achieve improved results, random forest was further optimized with the help of grid search using 45 different parameters, including the number of trees and the maximum depth. This enabled us to determine the most suitable settings to make right and consistent predictions. The output of Random Forest is further refined using three rule-based layers in order to increase the model predictions. The former is scam phrase recognition and identifies more than 60 known scam phrases. This functionality is context-sensitive to avoid misclassification of legitimate statements by analysing the presence of negation phrases like "no", "not" or "free" within a certain distance. Each phrase will contribute a risk value to the risk score of the fraud risk, up to a limit. The second element is red flag detection using six regular expression patterns with good designs. These patterns are typical of red flags such as free domain email address, payment deadlines, implicit or explicit threat or coercion to pay, mandatory payment, requesting deposits, personal phone numbers and non-corporate email IDs. Each red flag increases the score of the risk, up to a limit. The third layer involves the use of a true signal reducing component so as to reduce false alarms. When other elements in the email such as we do not charge, no registration fee, etc. and there are no strong indicators that the email is a fraudulent email, the probability of the email being a fake email is limited. This safeguards against false positives on genuine emails.

F. Module 3 - Website Legitimacy Check

This module assesses the link authenticity of websites cited in the email. Links are identified with a regular expression and the first link is inspected. WHOIS information is used to determine the age of the domain, while SSL certificate information is used to check for encryption and authenticity. A risk assessment is done based on these factors, according to thresholds described in Table II below.

Table II: Website Analysis Scoring Thresholds

Check	Condition	Risk
Domain age	< 30 days	+50 pts
Domain age	30–180 days	+25 pts
Domain age	180–365 days	+10 pts
Domain age	Undetermined	+20 pts
SSL cert.	Invalid / absent	+35 pts
SSL validity	< 60 days	+20 pts
SSL issuer	Unknown	+10 pts

G. Decision Engine

The InternGuard processing consists of a weighted decision engine in order to decide on the final classification based on the three modules. The total risk score is computed using a linear combination of all risk scores:

S = H \times 0.35 + C \times 0.45 + W \times 0.20

where S is the risk score and H, C, and W are the scores of the header, content and web (all in a range between 0 and 100). Weights are determined by the strength and usefulness of each of the modules. The content analysis module is highly weighted (45) as its weight is always on whereas the header module (35) yields reliable results when the provided data is in the form of .eml. The website module (20%) is relevant in the presence of a web link.

To make it strong, a safety net will be used to prevent the unintentional loss of valuable signals. When the content module includes scam words and has a high score (C > 60) the overall score will be raised to no

less than 55. This makes sure that the noticeable scam emails are not identified as authentic through the lack of other input. The last step is the determination of a threshold: the emails, where the final score S exceeds 52, are considered to be fake and the others are legitimate.

IV. RUNTIME IMPLEMENTATION

The system architecture is stratified, with stratification focusing on an individual aspect of the process.
The input layer will be implemented as a Flask-based web application that will enable the user to pass email text or upload email files in the eml format. It also features a dark mode and an easy-to-use four-step progress bar, which also aligns with the steps of the processing.

The file management layer then processes such files by temporarily storing them, and securely to avoid system-wide issues such as a lack of files because of locking. They are erased after they are processed to give the best performance.

The three detection modules are implemented at the analysis layer. Header analysis is only done when the .eml files are available and analysis of all inputs is done. The content analysis is performed in case the symbols are detected, and the analysis of websites is performed in case the URLs are detected. Results are normalized, to ensure a uniform calculation.

The decision level then computes the weighted scores and performs other rules, including the safety boost. The level of risk (high, medium, low) is based on the final score and the indicators are combined to give a general explanation.

Lastly, the presentation layer makes the results available to the user as a JSON. The user interface shows an animated risk indicator and the scores in each of the modules with explanations classified as key, primary or other; helping the user to know not only the outcome but also why.

A. System Flow of InternGuard

The whole cycle of InternGuard - how user input is taken through to the final output given to the user - is shown in Figure II.

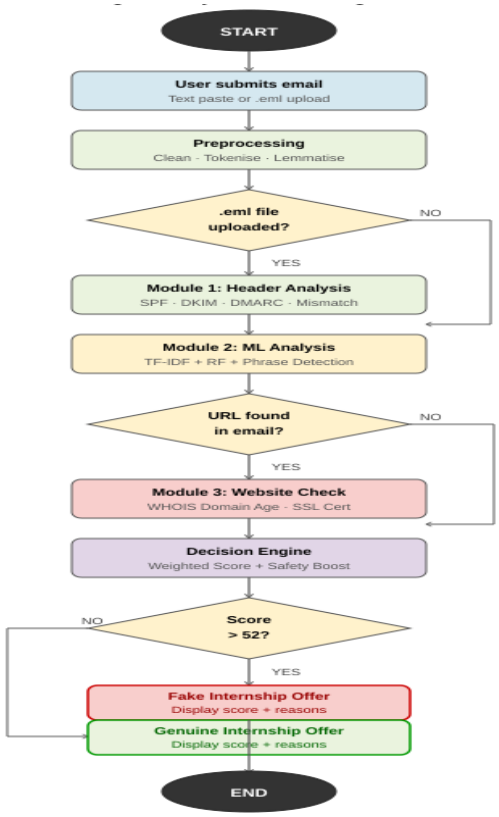

Figure II: System Flow of InternGuard

B. Technology Stack

Table III: Technology Stack

Component	Technology / Version
Language	Python 3.8+
Web Framework	Flask 2.x
ML Library	scikit-learn 1.2+
NLP	NLTK 3.8 (stopwords, WordNet)
WHOIS Lookup	python-whois 0.8
SSL Check	ssl + socket (stdlib)
Serialisation	joblib 1.2+
Data Handling	pandas 2.0 + numpy 1.24
UI	HTML5 + CSS3 + Vanilla JS
Development	Visual Studio Code

C. Hardware Requirements

Table III: Hardware Requirements

Component	Minimum	Recommended
Processor	Intel Core i3	Intel Core i5 or higher
RAM	4 GB	8 GB
Storage	5 GB free	10 GB free
OS	Windows 10 64-bit	Windows 10/11 64-bit

Network	Required	Broadband (WHOIS/SSL)
Browser	Chrome / Firefox	Chrome 90+ / Firefox 88+

V. RESULTS AND DISCUSSION

A. Algorithm Comparison

This set of 1,292 samples were used to evaluate and test 5 ML classifiers with a 5-fold stratified cross-validation approach. There were 10,000 features with trigram vectoriser, which did not change throughout all the comparisons. The test machine used for training was an Intel Core i7 (11th Gen CPU) with 8 GB RAM.

Table IV: Algorithm Comparison (*selected after tuning)

Algorithm	Acc %	F1 %
Logistic Regression	78.4	76.1
Linear SVM	80.2	78.9
Random Forest*	87.6	86.3
Gradient Boosting	85.9	84.7
Naive Bayes	72.3	70.8

Other algorithms were evaluated as well and the overall superior performance of Random Forest in obtaining higher F1-scores and receiver operating characteristic - area under the curve (ROC-AUC) scores made it the algorithm of choice. Although Gradient Boosting also had the high accuracy of 85.9%, it required nearly four times longer to train (151 seconds vs 40 seconds) which is not favorable on the side of iterative development and up-to-date information. In comparison, Naive Bayes the most accurate algorithm to-date at 72.3%; this is probably due to the fact that the independence assumption of features is not achieved in our correlated features (n-grams) in TF-IDF.

Once they had chosen the Random Forest as their model, they continued to use GridSearchCV and Stratified Fold (k=5) for further tuning. Fifty combinations of the different parameters were evaluated, including the number of trees, and the maximum depth of trees. A combination of 300 estimators, an infinity depth, was the most successful, achieving slightly better performance, with the F1-score rising to 86.3%.

B. Internal Evaluation Results

The resulting tuned Random Forest model was tested on the full sized 1,292-sample dataset. The confusion matrix is shown below in Figure 4 and ROC curve are shown below in Figure 5.2:

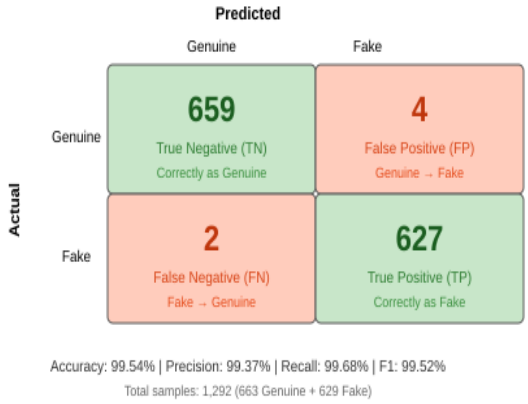

Figure III: Confusion Matrix — Internal Evaluation (1,292 samples)

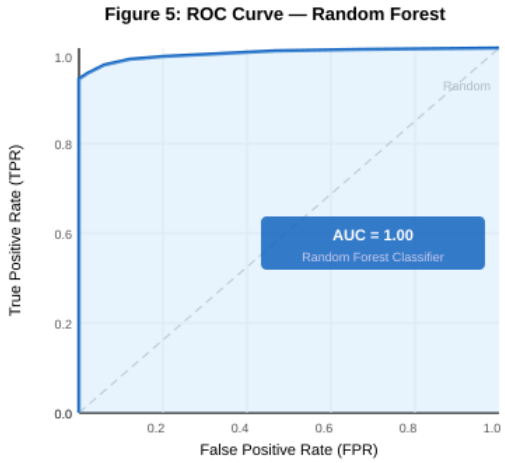

Figure IV: ROC Curve — Random Forest (AUC = 1.00)

The confusion matrix shows a very small number with only 4 false positives (genuine emails wrongly classified as fake) and 2 false negatives (fraudulent emails that were not considered as fake). Based on a closer examination of the false positives, it was found that these were legitimate business emails containing words like fee, registration and deadline used in a proper business context which caused their misclassification. However, false negatives were determined as work-from-home scam emails that

employed words indirectly and subtly. They did not directly state words like fee, but instead resort to more subtle terms such as contribution, and did not create a sense of urgency, making it less likely to be detected. These observations aided in realizing the limitations of the model and both of these kinds of errors were subsequently corrected by adding more targeted training samples.

C. Cross-Validation Analysis

Table V: 5-Fold Cross-Validation Results

Metric	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean
Accuracy	84.1	86.3	83.7	85.2	87.0	85.26 %
F1-Score	83.5	85.8	82.9	84.7	86.4	84.66 %
ROC-AUC	88.2	89.5	87.6	88.9	90.1	88.86 %

The low standard deviation in all splits of the data and the low standard deviation in the accuracy and F1-score (0.9 percent) demonstrates that the model is reliable in all splits of the data, and the model is not overfitted. Even though Fold 3 was slightly less accurate (83.7%), this is because borderline scam emails were randomly included in the validation set, thus making the classification more difficult.

D. External Validation

EMSCAD External validation was also conducted with 300 samples of the EMSCAD dataset (100 fraudulent and 200 natural job advertisements).

Table VI: External Validation on EMSCAD Dataset

Metric	InternGuard	EMSCAD Baseline [3]
Accuracy	56.33 %	89.0 % (in-domain)
Precision (fake)	41.62 %	—
Recall (fake)	77.00 %	—
F1-Score	54.04 %	—
True Positives	77 / 100	—
False Positives	108 / 200	—
False Negatives	23 / 100	—
Avg score — fake	61.6 %	—
Avg score — genuine	52.8 %	—

The cross-dataset analysis using the EMSCAD dataset was applied where the accuracy was 56.33% indicating the effect of the domain shift. InternGuard has been specially trained to recognize scam emails connected with internships in India where the emails often include such phrases as with pay and click to pay via UPI with free email addresses. The EMSCAD data on the other hand, consists predominantly of US and UK job ads which are variously structured and scammy. Nevertheless, the model still had a high recall of 77% of the fake data set and this shows that the model was able to correctly classify most of the scam emails. The occasional misclassification may be attributed to the varying character of language and structure of the emails in the two domains, which could end up raising suspicious flags on legitimate emails. But the evaluation shows that the fundamentals of email detection is effective even on a different dataset, indicating the model's ability to adapt to new data.

E. Real-World Email Testing

We also tested two real-world fake internship emails received by some of the MGIT students to further assess our system.

Table VII: Real-World Fake Email Test Results

Detail	SmartHire	Horizon Edge
Classification	Fake	Fake
Risk Score	100 %	100 %
Scam Phrases Found	7	8
Red Flags Found	6	2
Total Reasons	13	10
Key Phrase	security deposit	onboarding charge
Email Domain	@gmail.com	@gmail.com
Payment Amount	Rs 3,000	Rs 2,500

To further evaluate the practical effectiveness of InternGuard, two real-world fraudulent internship emails collected from MGIT students were analyzed. This experiment was conducted to determine whether the system could successfully identify scam indicators outside the training dataset.

InternGuard classified both emails as fraudulent with a risk score of 100%. The high scores were not triggered by a single feature alone, but rather by the

combined presence of multiple suspicious indicators, including scam-related phrases, payment requests, free email providers, and coercive language patterns.

In the SmartHire email, the system detected seven scam phrases and six additional red flags, resulting in a total of thirteen risk indicators. The phrase "security deposit" contributed significantly to the overall score because it is commonly associated with fraudulent recruitment schemes requesting upfront payments from candidates.

Similarly, the Horizon Edge email contained multiple suspicious indicators, including the phrase "onboarding charge," which represents another common variation of fraudulent payment requests. Both emails originated from free email providers rather than official corporate domains, further increasing the risk score.

These observations demonstrate that InternGuard is capable of identifying real-world internship scams through the combined analysis of linguistic, structural, and authentication-based indicators. The explainable output generated by the system also helps users understand why an email was classified as suspicious, making the detection process more transparent and user-friendly.

F. Comparison with Existing Systems

Table VIII: Comparison with Existing Systems

Feature	InternGuard	SpamAssassin	Google SB	Gen. ML
Internship-specific	Yes	No	No	Partial
Header forensics	Yes	Yes	No	No
ML content (88.52%)	Yes	No	No	Yes
Website age check	Yes	No	Yes	No
Explainable output	Yes	No	No	No
Indian scam patterns	Yes	No	No	No
Free email detection	Yes	No	No	No
Accessible web UI	Yes	No	Yes	Varies

The table above benchmarks InternGuard against three commonly used alternatives SpamAssassin, Google Safe Browsing, and general-purpose machine learning classifiers across eight capability dimensions.

The most striking difference is breadth. While each competing tool excels in one narrow area, none of them comes close to covering the full picture. SpamAssassin performs header forensics but stops there — it has no awareness of content semantics, no website verification, and no ability to explain its decisions to the user. Google Safe Browsing checks URLs against a known-malicious database and offers a web interface, but it operates entirely blind to email headers, message content, and the contextual red flags that make internship scams distinctive. General-purpose ML classifiers can learn from email content, but they are trained on broad spam categories and carry no specific knowledge of how fake internship offers are structured or worded particularly those targeting students in India.

InternGuard is the only system in this comparison that scores positively across all eight dimensions. Three of those dimensions internship-specific detection, Indian scam pattern recognition, and free email provider flagging — are completely absent from every alternative. These aren't minor edge cases. They represent the exact attack surface that internship fraudsters reliably exploit: posing as HR recruiters, sending offers from Gmail or Yahoo addresses, and mimicking the language patterns of legitimate Indian corporate communication.

The explainability gap is equally significant. Every other tool in this comparison produces a binary output spam or not spam with no reasoning attached. InternGuard surfaces the specific signals that contributed to its risk score, giving users the context they need to make an informed judgment rather than simply accepting or overriding a black-box decision.

Taken together, the table makes a clear case: existing tools were not designed with this threat model in mind. InternGuard was.

VI. CONCLUSION

This paper presented InternGuard, a multi-module intelligent framework for detecting fraudulent internship offer emails targeting undergraduate

students. The proposed system combines email header forensic analysis, machine learning-based content classification, and website legitimacy verification within a unified risk-scoring architecture.

Experimental evaluation demonstrated that the Random Forest classifier achieved the best overall performance among the evaluated models. The system obtained strong classification performance on the internal dataset and showed reasonable generalization capability during external validation using the EMSCAD dataset. In addition, testing with real-world internship scam emails confirmed the practical effectiveness of the proposed approach in identifying multiple independent indicators of fraud.

A major contribution of InternGuard is its explainable analysis mechanism, which provides users with detailed reasons behind each prediction instead of generating only a binary classification result. This improves transparency and helps students better recognize common recruitment scam patterns.

Future work will focus on improving cross-domain adaptability, incorporating transformer-based language models, supporting multilingual scam detection, and deploying the system as a scalable public web platform for broader accessibility.

REFERENCES

[1] P. Bhowmick and A. Hazra, "Comparative study of ML algorithms for spam detection," *Journal of Artificial Intelligence and Technology*, vol. 4, pp. 56–67, 2022.

[2] Salloum, T. Al-Qirim, and N. Abdallah, "Phishing email detection using NLP," *Procedia Computer Science*, vol. 189, pp. 175–182, 2022.

[3] Ilias, I. Roussaki, and S. Papavassiliou, "Detecting employment scams using EMSCAD," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 9, pp. 4012–4025, 2021.

[4] T. Alghamdi and I. Khan, "Fake job recruitment detection using machine learning," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 1234–1245, 2023.

[5] M. Kucherawy and E. Zwicky, "RFC 7489: DMARC," Internet Engineering Task Force, 2021.

[6] S. R. R. K. Sharan and R. M. S. Parvathi, "An AI-based real-time phishing detection system using machine learning," *SSRN Electronic Journal*, 2024.

[7] B. S. Kumar and P. R. Devi, "Phishing website detection using machine learning techniques," *International Research Journal of Modernization in Engineering Technology and Science (IRJMETS)*, vol. 7, no. 10, 2025.

[8] S. Nakamura, "Email sender verification using header field analysis," *IEICE Transactions on Communications*, vol. E105-B, no. 3, pp. 312–321, 2022.

[9] R. M. Mohammad, F. Thabtah, and L. McCluskey, "Domain age as phishing indicator," *International Journal of Computer Applications*, vol. 150, pp. 1–9, 2020.

[10] Guo, Y. Liu, and T. Wan, "BERT-based phishing email detection," in *Proc. AAAI Conference on Artificial Intelligence*, 2023, pp. 5891–5898.

[11] M. Ribeiro, S. Singh, and C. Guestrin, "Explainable AI in fraud detection," in *Proc. Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021, pp. 98–108.

[12] Apache SpamAssassin Public Corpus. Available: Apache SpamAssassin Public Corpus

[13] Real or Fake Fake Job Posting Prediction Dataset, Kaggle. Available: Kaggle Fake Job Posting Dataset

[14] Pedregosa *et al*., "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[15] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[16] K. Vishwanath, "Social engineering attacks targeting students," *IEEE Security & Privacy*, vol. 20, no. 4, pp. 45–53, 2022.

[17] S. Aggarwal, "Feature engineering for text classification," *Machine Learning Review*, vol. 12, pp. 1–28, 2021.