

Phishing Email Detection Using Ensemble Learning

Guilherme E. Kuglin¹, Ricardo de la Rocha Ladeira¹

¹Instituto Federal Catarinense (IFC)
Campus Blumenau
Blumenau – SC – Brazil

kuglin.guilherme@gmail.com, ricardo.ladeira@ifc.edu.br

Abstract. *Phishing attacks are a cybersecurity threat, especially via emails designed to deceive users and steal sensitive data. This work proposes a modular architecture for phishing e-mail detection in which content, URL features, and e-mail metadata are handled by independent specialized models, integrated at the decision level through stacking with out-of-fold meta-features. The main contribution is the modular design and the analysis of channel complementarity, rather than stacking itself. Over five independent runs, the ensemble reaches a macro F1-score of 0.9750, above the best individual model (0.9690), while ablation and error-overlap analyses provide evidence that the URL and metadata channels correct errors made by the textual channel.*

1. Introduction

Phishing attacks continue to be among the primary threats to digital security, being widely used to obtain credentials, financial information, and sensitive data from users and organizations. Recent security reports indicate a continuous growth in the number of targeted campaigns [Anti-Phishing Working Group 2024]. This attack type uses Social Engineering to deceive victims into visiting malicious pages or executing dangerous files via fraudulent communications impersonating legitimate entities [Jakobsson and Myers 2006].

Due to the increase in the volume and sophistication of these attacks, traditional filtering approaches based on blocklists, signatures, or static rules present limitations in identifying newer campaigns and previously unknown variants. In this context, machine learning techniques have been widely explored for the automatic detection of phishing [Gualberto et al. 2020, Cruz 2025]. However, many of these approaches focus on a single source of information, such as the text body of the message or URLs present in the email, reducing the system’s generalizability against different strategies used by attackers.

This paper proposes a phishing detection architecture based on ensemble learning [Dietterich 2000], integrating specialized models to analyze content, URL characteristics, and email metadata. The approach is modular: it uses independent models for each feature group and combines their outputs through a stacking strategy ¹. The objective of this work is to evaluate an architecture capable of exploring multiple sources of information in emails, verifying the individual performance of each feature group, how much each contributes to the system’s final decision, and whether they provide complementary information. The main contribution lies in this modularity and in the analysis of the complementarity among channels, rather than in the use of stacking itself.

¹Stacking is an ensemble learning technique in which the outputs of multiple base models are used as input for a meta-classifier responsible for the final decision.

2. Background and Related Work

Ensemble learning involves combining multiple machine learning models to improve the classification system's generalization capability and robustness, enabling the integration of diverse analytical perspectives based on complementary features found in the data [Dietterich 2000]. Among the strategies involved, the stacking method is particularly relevant in phishing detection scenarios, where different components of an email may exhibit different patterns of malicious behavior.

In the literature, approaches to phishing detection typically prioritize a single group of features. Studies focusing on textual content explore representations of message subjects and bodies [Gualberto et al. 2020], while recent work reports promising results with ensemble methods, including hybrid ensembles designed specifically for phishing e-mails [Bountakas and Xenakis 2023, Cruz 2025]. The study presenting the MeAJOR dataset [Alvarado et al. 2025] also evaluates various combinations of text, URL, and attachment features, achieving its best results with single models trained on aggregated features. The proposed architecture differs from these approaches by maintaining each feature group in a separate model and integrating them only at the decision stage. This allows for the evaluation of each group's isolated performance and an assessment of the weight assigned to each within the final classification.

3. Method

The proposed architecture was designed to automatically detect *phishing* emails through the joint analysis of different characteristics extracted from the messages. Textual content, attachment/language metadata, and URL attributes were analyzed, as phishing campaigns often exploit multiple elements to simulate legitimate communications. The approach uses three specialized models (Content, URLs and Metadata), integrating their outputs through ensemble learning based on stacking.

3.1. Dataset and Preprocessing

For training and evaluating the models, the MeAJOR dataset [Alvarado et al. 2025] was used, chosen due to its variety of attributes, which allows exploring multiple analysis perspectives. MeAJOR includes information such as subject, email body, number of URLs, domain characteristics and information about attachments. After removing a record that had an invalid label, the set used contained 108,684 emails, 60,650 of which were legitimate (55.8%) and 48,034 were malicious (44.2%). This data was divided into training and test sets in the proportion of 80% and 20%, stratified by class, which resulted in 86,947 emails for training and 21,737 for testing, preserving the original distribution of classes in both sets and in all executions performed. As no hyperparameter search was performed, the test set was used exclusively for the final evaluation of the models.

In the preprocessing stage, the subject and body fields were concatenated into a single representation of the email content. Text normalization, including lowercase conversion and tokenization, was performed directly by the vectorizer, without applying stemming, lemmatization, or stopword removal. MeAJOR natively provides texts with anonymized entities via tokens such as [NAME] and [URL], which is a dataset characteristic rather than a preprocessing step. URL-based features (e.g., total link count, average URL length, and subdomain count) and metadata (e.g., attachment presence/types

and message language) were also extracted. Duplicate messages were not removed prior to data splitting. 3,751 identical subject-body pairs were identified (3.5% of the dataset), which may introduce data leakage between subsets. This limitation may yield overly optimistic estimates for the Content Model and overestimate the reliability of textual meta-features across cross-validation folds, potentially inflating the weight assigned to the content channel in the meta-classifier, an effect not quantified in this work.

3.2. Content Model

Content analysis applied Natural Language Processing techniques to the subject and body of the emails, following normalization and text preparation steps for supervised classification models. An approach based on Term Frequency-Inverse Document Frequency [Salton and Buckley 1988] was used, limited to 5,000 terms and restricted to unigrams, combined with Logistic Regression [Hosmer et al. 2013] in its default configuration, using the `lbfgs` solver and a regularization parameter of $C=1.0$, selected for its low computational cost and strong preliminary performance during experiments.

3.3. URL Model

The URL analysis module was developed to identify structural patterns frequently associated with phishing campaigns, using the number of URLs, average link length, and number of subdomains. These features were selected because phishing campaigns commonly employ long URLs and multiple subdomain levels to complicate the detection of malicious pages by users. It is worth noting that the attributes used are strictly structural, without incorporating domain reputation or similarity analysis. The extracted features were used to train a Random Forest model [Breiman 2001], consisting of 100 trees with no depth limit, chosen for its ability to handle multiple structural attributes and identify non-linear relationships among the analyzed variables.

3.4. Metadata Model

The Metadata Model uses information present in email metadata, including the presence of attachments, content types, general information related to message structure, and the identified language, as these represent complementary information to textual content and URLs. Since language is a categorical variable, it was numerically encoded based on the training set and applied unchanged to the test set, assigning an unknown category code to languages not observed during training. As with the URL module, classification was performed using a Random Forest model [Breiman 2001] with 100 trees.

It is worth noting that header-based inconsistency analysis (e.g., `From/Reply-To`, `Return-Path`, `Received chain`, `SPF`, `DKIM`, and `DMARC`) is outside the scope of this module, since such information is not available in the dataset.

3.5. Design and Evaluation of the Proposed Ensemble

The models developed for content, metadata, and URL analysis were integrated using ensemble learning based on stacking [Dietterich 2000]. In this strategy, each model generates a classification probability for the analyzed email, serving as input for a Logistic Regression meta-classifier responsible for the final decision. Thus, the proposed architecture (Figure 1) seeks to combine different analytical approaches, enhancing detection robustness in scenarios involving diverse phishing strategies.

To train the meta-classifier, the base model probabilities were obtained using five-fold cross-validation, ensuring that the probability assigned to each email in the training set is always produced by an out-of-fold model that did not see that sample during training. This procedure prevents the meta-classifier from fitting to optimistic estimates from the base models, reducing the risk of data leakage and overestimation of confidence assigned to models with a higher tendency to memorize the training set.

Evaluation was conducted using accuracy and F1-score metrics, considering a classification correct when an email was properly identified as legitimate or malicious. The F1-score is reported using macro-averaging, calculated separately for each class and then averaged without frequency weighting, treating the legitimate and malicious classes with equal importance. The entire experiment was repeated across five distinct random seeds, with results reported as the mean and standard deviation of the runs. The source code, training scripts, and complementary materials are available in a public repository².

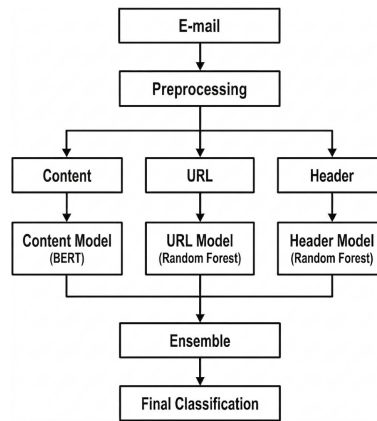


Figure 1. Phishing detection architecture using ensemble learning.

4. Results

The experiments individually evaluated the performance of the Content, URL, and Metadata models, as well as selected combinations and the final architecture. Table 1 shows the mean and standard deviation on the test set across five runs.

Table 1. Results obtained on the test set (average $\pm$ standard deviation, 5 runs).

Model	Accuracy	F1-score (macro)
Content Model	0.9694 $\pm$ 0.0009	0.9690 $\pm$ 0.0009
URL Model	0.8446 $\pm$ 0.0021	0.8399 $\pm$ 0.0022
Metadata Model	0.7241 $\pm$ 0.0033	0.6924 $\pm$ 0.0044
Content + URL	0.9744 $\pm$ 0.0011	0.9741 $\pm$ 0.0011
Content + Metadata	0.9706 $\pm$ 0.0012	0.9702 $\pm$ 0.0012
Proposed Ensemble	0.9754 $\pm$ 0.0011	0.9750 $\pm$ 0.0011

The results demonstrate that the Content Model achieved the best individual performance, highlighting the relevance of textual features in the email subject and body for identifying linguistic patterns typical of Social Engineering. The URL Model showed intermediate performance, whereas the Metadata Model yielded the lowest result, although all outperformed the majority class baseline (0.558 accuracy and approximately 0.358

²<https://github.com/Kuglin/PhishingDetection/>

macro F1-score). In isolation, the Metadata Model is conservative toward the malicious class (0.85 precision, 0.45 recall), misclassifying few legitimate emails but failing to detect over half of the phishing messages.

The ensemble learning-based architecture achieved the best overall performance, with a macro F1-score of 0.9750 ± 0.0011 compared to 0.9690 ± 0.0009 for the standalone Content Model. Across five independent runs, the proposed ensemble consistently achieved higher macro F1-score than the Content Model. However, the absolute gain is small (0.0060 in macro F1-score) and no statistical test was applied. Given the limited number of runs, these results should not be interpreted as statistically significant. In relative terms, the gain represents a reduction of approximately 20% in the error rate of the Content Model (from 3.06% to 2.46%, or about 130 fewer misclassified emails in the test set). The weights learned by the meta-classifier (the Logistic Regression coefficients, directly comparable as all inputs are probabilities on the same $[0, 1]$ scale) were 9.08 ± 0.04 for the content channel, 3.72 ± 0.05 for URLs, and 2.34 ± 0.04 for metadata. These values proved stable across executions, with a standard deviation below 0.5% of the mean value for all channels, reinforcing that the observed hierarchy among channels is not an artifact of a specific split, although it does not rule out the effect of the duplicates.

The ablation results in Table 1 show that adding URL or metadata information to the Content Model provides additional predictive information. The contribution of the URL channel (0.0051 macro F1-score) exceeds that of the metadata channel (0.0012), consistent with the weights assigned by the meta-classifier. Combining all three channels outperforms any evaluated subset, indicating that the modules provide complementary rather than redundant information.

The complementarity between channels is also directly observable in the overlap of classifier errors. On average, the Content Model misclassifies 664.6 emails in the test set, of which only 104.0 ± 9.9 are also misclassified by the other two modules. In other words, in 560.6 cases, representing 84.4% of the Content Model’s errors and 2.6% of the test set, at least one of the other channels assigns the correct classification. Evaluated separately, the URL Model corrects 443.4 ± 13.5 of these errors, while the Metadata Model corrects 380.0 ± 8.7 . These results provide experimental evidence that structural and metadata channels correct errors made by textual analysis rather than merely reinforcing decisions it would already make correctly.

Compared to the top result reported for MeAJOR [Alvarado et al. 2025] (0.9834 F1-score via XGBoost on text and URL features), the proposed architecture shows lower absolute metrics. However, direct comparisons require caution due to potential differences in evaluation protocols. Moreover, the proposed design prioritizes channel modularity over maximizing performance via concatenated features, enabling independent channel evaluation, weight inspection, and component updates without full retraining.

5. Conclusion

This work presented a modular architecture for phishing email detection, where content, URL, and metadata channels are independent components integrated through stacking with out-of-fold meta-features, and analyzed the complementarity among these channels.

The results indicate that textual content concentrates most of the observed discriminative power, while the URL and metadata channels exhibit lower individual perfor-

mance, yet remain superior to majority class assignment. The proposed ensemble outperformed the best standalone model in all five independent runs. The ablation analysis shows incremental performance gains when adding each channel to the Content Model, while the error overlap analysis indicates that these channels correct a substantial portion of its errors, confirming feature complementarity.

As next steps, we plan to: (i) repeat the experiments after removing duplicate and near-duplicate messages, or using grouped splits, to quantify the leakage impact on the Content Model and on meta-classifier weights; (ii) add a baseline trained on concatenated features and paired statistical tests over more runs, to assess gains over the Content Model; and (iii) evaluate Transformer-based models, mainly BERT (Bidirectional Encoder Representations from Transformers) [Devlin et al. 2019], additional content and attachment features, and more heterogeneous datasets, including sources unseen during training.

Acknowledgements

The authors declare that Gemini 3.6 Flash was used in this work for Portuguese-to-English translation, L^AT_EX formatting, and editing Figure 1.

References

- Alvarado, E. et al. (2025). Meajor: A multi-feature email dataset for advanced junk, phishing, and organizational risk detection. *arXiv preprint arXiv:2507.17978*.
- Anti-Phishing Working Group (2024). Phishing activity trends report. <https://apwg.org/trendsreports/>. Accessed on: May 21, 2026.
- Bountakas, P. and Xenakis, C. (2023). HELPHED: Hybrid ensemble learning phishing email detection. *Journal of Network and Computer Applications*, 210:103545.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.
- Cruz, L. P. (2025). Detecção de phishing por aprendizado de máquina com ênfase em métodos ensemble. Trabalho acadêmico.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*.
- Dietterich, T. G. (2000). Ensemble methods in machine learning. *International Workshop on Multiple Classifier Systems*.
- Gualberto, E. S., De Sousa, R. T., Thiago, P. D. B., Da Costa, J. P. C., and Duque, C. G. (2020). From feature engineering and topics models to enhanced prediction rates in phishing detection. *IEEE Access*, 8:76368–76385.
- Hosmer, D. W., Lemeshow, S., and Sturdivant, R. X. (2013). *Applied Logistic Regression*. Wiley.
- Jakobsson, M. and Myers, S. (2006). *Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft*. Wiley.
- Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523.