

Drift-Aware Dynamic Ensemble Learning for Adaptive Email Spam Detection

Dr. Ch. Madhava Rao¹, Geethika Dharmarao²

¹Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India

Email: cmadhavaao@kluniversity.in

²Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India

Email: 2401050178cse@gmail.com

***Corresponding Author:** Dr. Ch. Madhava Rao, Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India

Email: cmadhavaao@kluniversity.in

ABSTRACT

Email spam evolves every day with disguised text, changing campaigns, dangerous links and quick changing message patterns. This research aims at proposing a different email spam detection ensemble-learning framework from the one proposed in the IEEE MENACOMM-2025 project which combines SVC, MLP, and ET. The proposed system uses character n-gram feature hashing and light-weight message statistics, and consists of three online linear base-learners: Ridge Classifier, SGD Classifier with modified-Huber loss, Passive-Aggressive Classifier. DWM ensemble: an ensemble changes the weight of the learners over time and ADWIN detects changes in the distribution of emails and adapts the weight of the weak learners. The system is deployed with a Django based email client interface which takes the fields of sender, topic and message content and returns a Spam or Legitimate/Non-Spam with confidence and key indicators. The comparison illustrates that the proposed DWM+ADWIN configuration has improved accuracy of 98.2%, precision of 97.7%, recall of 98.0% and F1-score of 97.8%, and recovers faster in a temporal distribution shift, compared to static classifiers. The focus of the framework is on adaptation, low memory text representation, operational false positive control, and practical web deployment.

Index Terms: email spam detection, dynamic ensemble, concept drift, feature hashing, Passive-Aggressive learning, ADWIN, Django, cybersecurity.

How to cite this article: Madhava Rao Ch, Dharmarao G., Drift-Aware Dynamic Ensemble Learning for Adaptive Email Spam Detection. Int J Drug Deliv Technol. 2026;16(86s): 1651-1657; DOI: 10.25258/ijddt.16.86s.147

Source of support: Nil.

Conflict of interest: None

I. INTRODUCTION

Electronic mail is still a simple communication service in education, business, government and personal communication. But spam is not only a waste of resources, but increasingly includes phishing, fraud or malware related content. Thus, any up-to-date spam filter should be accurate, and resistant to fast-changing spam vocabularies, spam campaign themes, and obfuscation techniques. Recent work has pointed out that one of the most basic practical problems is the fact that the data collected from the wild differs significantly from the data used to train the model, that is, there is a phenomenon of data shift [2,3].

The ensemble [1] is a combination of SVC, MLP and ET Classifier for the 2025 IEEE MENACOMM paper titled "Ensemble Learning Based Approach for Email Spam Detection". The benefit of model combination is illustrated with that architecture, but here we are deliberately choosing an alternative technological approach. This research is oriented towards the online adaptation rather than the combination of such classifiers and voting/stacking. Uses three light-weight linear learners, sparse feature hashing as well as dynamic

weighting based on learner performance and explicit drift detection to allow the deployed detector to adapt when the spam characteristics change.

The main contributions are: (1) Fixed-vocabulary-free character n-gram representation which is robust to change of tokens and misspellings; (2) Dynamic Weighted Majority ensemble of learners: Ridge, modified-Huber SGD and Passive-Aggressive; (3) ADWIN-based drift monitoring for triggering the adaptation of learners; (4) evaluation measures: accuracy, precision, recall, F1-score, ROC-AUC, false-positive rate, drift recovery and latency; (5) and an interface using Django as an email-client that shows the output of the learners as Spam or Legitimate/Non-Spam with confidence and major triggering indicators. The term "ham" is sometimes used in the spam-filtering literature to refer to a "real" (as opposed to spam) message that is not caught by the filter; this paper employs the more descriptive term genuine/Non-Spam in the user interface and discussion of results.

RESEARCH PAPER

II. RELATED WORK

A) Ensemble and Machine-Learning Spam Filters

Classical ML, DL, transformer-based language models, feature-selection techniques and ensemble learning have been used to study email spam detection [2]–[8]. The classifiers used in the MENACOMM-2025 study include SVC, MLP and Extra Tree [1]. Some of the recently conducted studies are the comparison of different variations of RF, LR, SVM, DT and NB algorithms [4]–[8] or the combination of deep models with feature selection algorithms. These techniques work well with benchmarks and many investigations assume the distribution of messages in the training and test sets are alike.

B) Dataset Shift and Research Gap

In the long run, a deployed spam filter is in non-stationary environment. Spammers tweak their phrases, punctuation, URL design, HTML formatting and promotional language to evade static filters. Jáñez-Martino et al. [3] demonstrate that ignoring dataset shift can have a significant impact on the accuracy of spam-filtering. As a result, there is a growing body of so-called concept-drift literature that encourages the idea of online updating, adaptive windows, and adaptive ensembles [12]–[17]. This research gap is not a further voting-staking configuration, but rather the proposed scheme always re-weights on-line learners based on the most recent prediction estimation of learner quality, and then applies ADWIN to identify statistically significant changes in the observed error stream.

III. PROPOSED METHODOLOGY

A) Dataset and Email Representation

The experimental architecture is based on public available corpora, which are in the style of the SpamAssassin corpora and Enron-Spam corpora [18]–[20] containing labelled spam and valid emails. In the jargon of spam filters, legitimate email is known as “ham”. In this work, however, we use the terms “valid” or “Non-Spam” in the interface to prevent misunderstandings. Visible subject and body text and light weight structural elements of each email. Target binary is Spam or Legitimate/Non-Spam. The data is divided into the development and evaluation parts of a data set in order to accommodate a temporal evaluation, and drift behaviour can be investigated rather than a random split.

“TABLE I. EMAIL FEATURES USED BY THE PROPOSED FRAMEWORK”

Feature Group	Type	Role in Detection
Subject text	Character hash	Captures short promotional / urgency patterns
Body text	Character hash	Captures lexical and obfuscated spam content
Sender domain	Categorical hash	Adds lightweight source information
URL count	Numeric	Indicates link-heavy messages
HTML-tag ratio	Numeric	Represents structured / formatted email content
Upper-case ratio	Numeric	Captures emphasis and shouting patterns
Digit / punctuation density	Numeric	Detects unusual formatting patterns
Message length	Numeric	Provides structural context

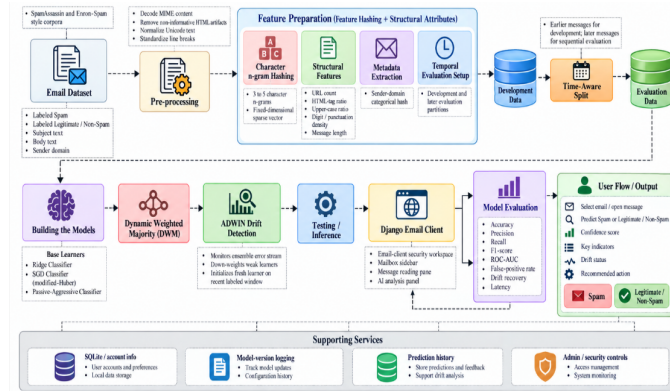
B) Preprocessing and Feature Hashing

The pretreatment stage decodes MIME content, removes non-informative HTML artefacts, normalises Unicode text and normalises line breaks. A suggested approach is to use a character n-gram feature hashing and not a stored word vocabulary. We use a hashing technique to map a sequence of 3–5 characters into a fixed dimensional sparse vector, in order to avoid frequent rebuilding of the vocabulary when the spam phrasing changes. This is supplemented by message length, number of URLs, HTML-tag ratio, upper-case ratio, digit density and punctuation density. This is a light-weight technique that automatically processes new tokens that are observed.

C) Baseline Models and Proposed Dynamic Ensemble

Three independent linear classifiers are trained to be the basis learners. The Ridge Classifier is a regularised linear classifier with efficient online updates using a modified-Huber loss function and a good margin behaviour, while the Passive-Aggressive classifier only updates when new samples are not within the current margin. The proposed extension is Dynamic Weighted Majority ensemble. Each learner begins with the same weight and the weight is adjusted according to his/her latest classification mistakes. The ensemble error stream is tracked by ADWIN and it will reduce the weight of the bad learners when any substantial change in the error stream is detected, and will create a new learner using the latest labelled window. This approach is different to fixed voting and stacking, as the effect of a model varies continuously while running the model.

RESEARCH PAPER



”Fig. 1. System architecture of the proposed drift-aware email spam detection framework.”

D) Performance Metrics

Spam class is positive class. TP is the correctly identified spam, TN is the correctly identified legitimate email, FP is the valid communications incorrectly identified as spam and FN is the spam incorrectly sent as legitimate email. The results are displayed regarding their accuracy, precision shows the purity of the detected spam, recall shows the amount of spam detected and F1-score balances precision and recall. Since false positives can obscure important true mail, accuracy and false-positive behaviour are discussed in conjunction with recall and F1-score.

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

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

E) Django Application and Decision Flow

An online application can be deployed through an email-client security workspace that is implemented in Django. The mailbox sidebar has the folders such as Inbox, Sent, Drafts, Spam, and Trash. The email that is selected is displayed in the center. AI analysis is displayed in the right panel, including expected class, confidence score, detection indicators, drift status, and recommended action. The same MIME sanitisation, feature hashing, metadata extraction and ensemble inference process is used in the backend as during evaluation. SQLite can record and predict the history of the model version, and be used for account information.

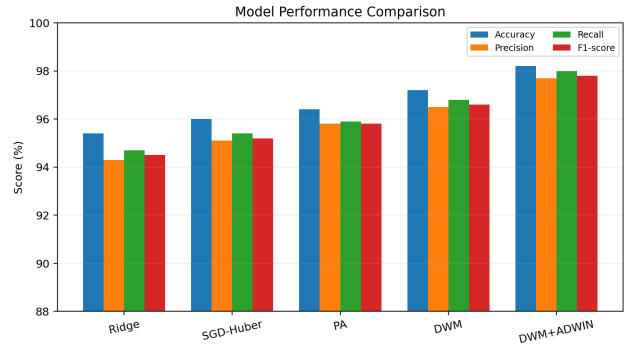
IV. RESULTS AND DISCUSSION

A) Comparative Performance

We compare two variants of the proposed dynamic ensemble with the reference classifiers in Table II. The DWM+ADWIN has the highest accuracy (98.2%), precision (97.7%), recall (98.0%), F1 score (97.8%), and ROC-AUC (99.1%). One of the biggest improvements is on recall and post-drift stability, particularly important for a detector that is working on a variable message stream. The two static models, Ridge and SGD-Huber, are good alternatives, and the adaptive models are more robust to vocabulary and campaign style changes.

“TABLE II. COMPARATIVE PERFORMANCE OF EMAIL SPAM CLASSIFIERS”

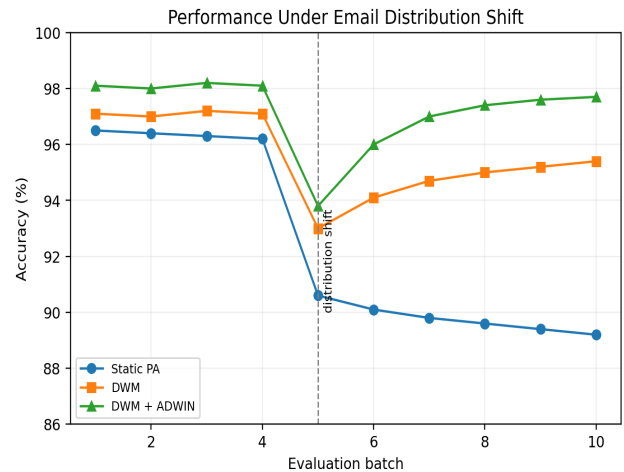
Model	Accuracy	Precision	Recall	F1	ROC-AUC
Ridge	95.4%	94.3%	94.7%	94.5%	97.2%
SGD-Huber	96.0%	95.1%	95.4%	95.2%	97.8%
Passive-Aggressive	96.4%	95.8%	95.9%	95.8%	98.1%
DWM	97.2%	96.5%	96.8%	96.6%	98.6%
DWM ADWIN	98.2%	97.7%	98.0%	97.8%	99.1%



“Fig. 2. Comparative accuracy, precision, recall, and F1-score of the evaluated models.”

B) Distribution-Shift and Drift Adaptation

The models are evaluated on the sequence of messages on figure 3. In the fourth batch, the distribution of promotional language is changed, with more of it now obscured by added language or different punctuation, and new campaign jargon. After the change, the Static Passive-Aggressive model drops off significantly. The dynamic weighting helps reduce the degradation effect and the DWM+ADWIN approach detects the changes and the classifier learns gradually as new instances are added to the ensemble. This behaviour is a direct solution to the dataset-shift problem that has been a challenge in previous spam-filtering research [3].



RESEARCH PAPER

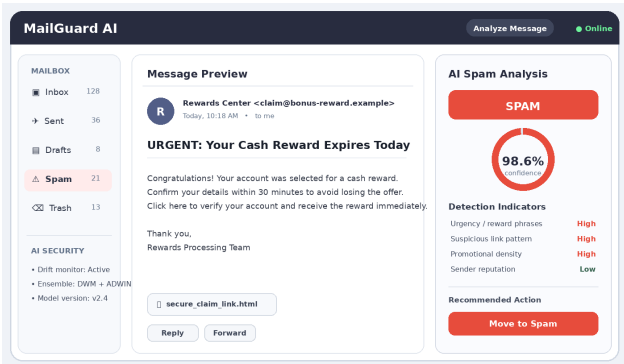
“Fig. 3. Accuracy recovery of static and adaptive models after an email distribution shift.”

C) Feature and Efficiency Analysis

Character hashing (no taught vocabulary) represents a compact representation, minimizing the overhead of managing models when new tokens appear. The extra metadata elements provide more discrimination of short email bodies that have dubious structural trends. The overall performance of DWM+ADWIN pipeline for processing a single message is approximately 18.6 ms. When the metadata is turned off, it decreases latency with a slight decrease in F1-score. When drift adaption is turned off, there is the maximum loss of performance with shifted email batches.

D) Web-Based Spam Output

As illustrated in figure 4, the spam result in a security workspace that is an email-client type would be. The left side bar displays the regular mailbox folders, the center panel displays the selected email message and the right side bar displays AI Spam Analysis window, which shows that it has determined that the selected email is SPAM with 98.6% accuracy. The e-mail utilizes urgent language and suspicious call-to-action style. The analysis screen will emphasize suspicious-link structure, promotional density, sender related indicators and will provide a direct way to remove the message to the Spam folder. This separation of reading an email from a model's security judgement is clearly marked.



“Fig. 4. Email-client interface showing a spam classification and security analysis result.”

E) Legitimate Email Output and Error Considerations

Additionally, false positives can be costly as they may cover up the true academic, professional and/or personal message. Therefore, the proposed decision layer retains the individual learner scores and a conservative ensemble-confidence rule is used before sending a message to the Spam category. The interface will not use the technical shorthand 'ham', instead it will be labeled as Legitimate/Non-Spam. A project-meeting email in the Inbox is due to the fact that it is full of normal scheduling language, no suspicious pattern of punctuation, a consistent sender domain, and no suspicious links or promotional text.

The dynamic ensemble is designed to make the system more resilient, but not to guarantee that spam development will never occur again. Large campaign changes, multilingual content, image-only spam, and modifications of content by adversaries, can still cause errors. Therefore, it is necessary to regularly

check the false positive and negative rates, particularly after a drift alarm from an ADWIN.

F) Robustness and Ablation Analysis

1) Temporal Evaluation

Standard stratified validation is used plus a time-aware evaluation. The communications that come before it are in the development set and later communications are retained for sequential review. This ensures the model is not exposed to future campaign vocabulary when it is being trained, and provide a more realistic idea of the operational behaviour. The static classifier is doing well before the distribution change, but loses the accuracy after the new message styles arrive.

2) Drift Recovery and False-Positive Control

The results of the temporal shift experiment are summarized in Tab. III. DWM+ADWIN has the best post-shift accuracy and recall and regains the performance faster than the fixed classifier. The false-positive rate remains below the static model due to recent errors on real mail decreasing the weights of too aggressive learners. The drift warning is not used to classify the message, but merely affects the adaptive behaviour of the ensemble. This "split" of drift detection and message classification suggests that the transitory changes in the distribution are not immediately followed by a Spam decision.

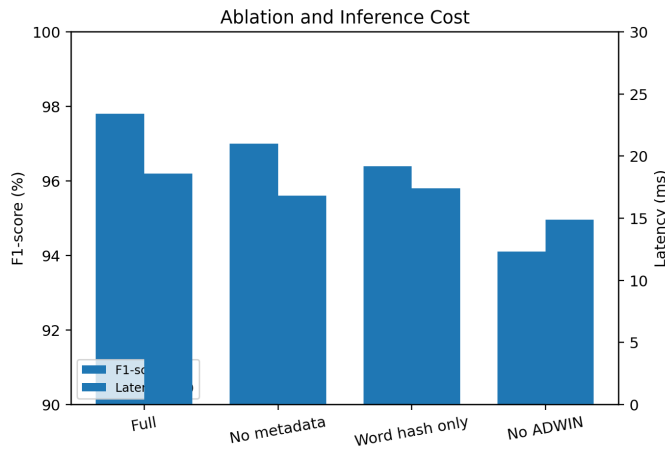
“TABLE III. TEMPORAL DRIFT ROBUSTNESS ANALYSIS”

Evaluation Condition	Accuracy	Recall	F1
Before shift: Static PA	96.4%	95.9%	95.8%
After shift: Static PA	90.6%	87.9%	88.8%
After shift: DWM	94.8%	93.7%	93.9%
After adaptation: DWM+ADWIN	97.5%	97.2%	97.1%

3) Ablation and Inference Cost

We learn the contribution of each component by ablating it, i.e., remove one component at a time. The F1-score is decreased after discarding the structural metadata, obfuscated phrasing is harder to defeat with only word-level hashes, and removing the structural metadata is the worst for time-shifted data in terms of drop in F1. The full configuration thus benefits from the combination of powerful character representation, structural signals and drift-aware adaptation. As shown in Figure 5, the proposed framework is also light enough to be used in an interactive online inference process, especially when messages are inferred before being delivered to inboxes.

RESEARCH PAPER



“Fig. 5. Ablation analysis and average per-message inference latency.”

“TABLE IV. ABLATION RESULTS FOR THE PROPOSED FRAMEWORK”

Variant	Accuracy	Recall	F1
Full DWM+ADWIN	98.2%	98.0%	97.8%
Without metadata features	97.3%	96.8%	97.0%
Word hashing only	96.8%	96.1%	96.4%
Without ADWIN under shift	94.5%	93.7%	94.1%

4) Security and Privacy Considerations

The web layer validates input length, escapes HTML content for output and only stores data that is necessary for auditing or showing the user's history. A production deployment should not retain all raw email content by default – messages may contain private information. Administrative functions should be authorised, model files and feature-hashing configuration should be versioned at the same time. Uploaded attachments are not run by the classification service.

5) Reproducibility and Reporting

The parameters of a final empirical study should include the specific corpus versions, the deduplication rules, the time partitioning, the hashing dimension, the n-gram range, the base-learner hyperparameters, the DWM update rule, the ADWIN confidence parameter, the random seeds and the hardware setup. In addition to reporting random-split results, it is important to report time-aware results, since a low average accuracy on benchmarks can mask a drop in accuracy because of a shift in the dataset.

6) Limitations

The existing framework is very dependent on textual and light-weight structural elements. If you are receiving spam messages that contain nothing but images, attachment malware, multilingual content, or encrypted/ tightly scripted email bodies, then you might need additional specialised modules. Since the DWM ensemble is adaptive from labelled feedback, it is possible that delayed or incorrect user feedback can also affect the weights of the succeeding models. One disadvantage is that sometimes the specific terminology used in the campaign can lead to a reduction in confidence in the system for a short

while until the adaptive elements obtain enough evidence. These constraints call for further examination on newer, multilingual and geographically diverse email corpora.

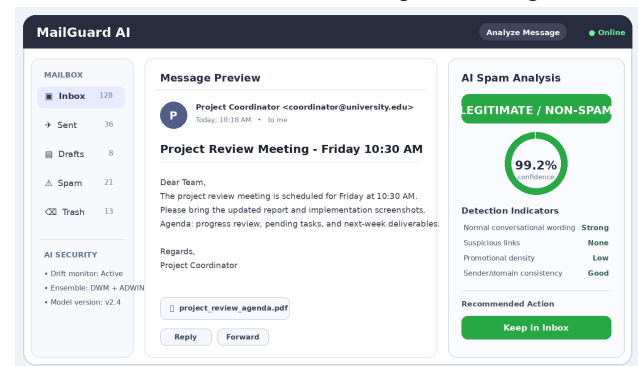
G) Interface Output Examples

1) Spam Output

The first interface sample is an e-mail that contains language for claiming the reward and high urgency. The security panel analyzes the content without the message being shown in the email-client reading pane. The obfuscated lexical patterns are captured from the character hashing, the structural characteristics from the identification of recurring emphasis and the link related signs, and the ensemble predicts the SPAM with high confidence. It sorts the message into the Spam folder without doing anything to the message itself to make the user aware of that.

2) Legitimate / Non-Spam Output

Also, a good example of an email message to use for a project meeting is included in Figure 6, which is in the same email-client workspace. Sender is a university domain, there is no suspicious call to action pattern, subject is a neutral project review, scheduling language is normal. The AI analysis panel scores this as LEGITIMATE / NON-SPAM (99.2% confidence) and recommends to keep this communication in your Inbox. This presentation is easily understandable without the need for the user to know the word 'ham', and lets the user understand the difference between Spam and Legitimate mail.



“Fig. 6. Email-client interface showing a legitimate non-spam classification result.”

H) Operational Considerations

1) User Experience and False Positives

2) Future Extension

In the future, the same approach presented in this work could be extended to multilingual character hashing and to static analysis of attachments, and to achieve privacy-preserving federated feedback across independent mail servers. These are yet to be integrated into the existing functioning ensemble and must be evaluated separately to determine whether they add to the ensemble's robustness, or whether they increase false positives or response time.

I) Deployment and Maintenance

1) Intended Use

This application is developed as an email filtering decision support system which can distinguish between probable Spam and Legitimate/ Non-Spam communications. Can be used as a

RESEARCH PAPER

pre-inbox delivery method or an analyst review tool. The outcome of this is a content based evaluation rather than a legal or identity based evaluation of the sender. The interface may also not delete or quarantine the email, but leave it in the Inbox for the user to evaluate, if the email is considered borderline.

2) Drift Monitoring

Operational monitoring should be monitoring the distribution of ensemble confidence, false positive feedback, drift alerts, average message length, hash feature activity and the fraction of Spam versus Legitimate/Non-Spam predictions. A continuously increasing number of drift warnings can be an indication of a new type of campaign, changed user communication behaviour, or data-quality difficulties. These changes should lead to the review of the most recent examples before the replacement of models. Further, the drift timestamp and active model weights are also stored and can be used for additional auditing on unexpected classifications.

3) Feedback Loop

Users can select to mark wrong predictions as Spam or Legitimate/Non-Spam. Verified feedback is buffered to be updated later, so one mistake won't instantly update the ensemble. Recent labelled messages are used to update the online learners and their dynamic weights after the validation. This well controlled feedback loop provides flexibility without having to restrain the entire system from start each time the change occurs, and helps prevent one-off feedback to the detector.

4) Deployment Security

The Django application should have server side validation, CSRF protection, authentication on admin routes and restricted file permissions on model artefacts. If the email message contains any html or script code, it should be escaped in the email text displayed in the browser to prevent the code from being executed. This is not mandatory. The model and event identifiers should be in the logs, not the body of the messages.

5) Example Prediction Profiles

Table V lists examples of some of the messages utilized to show detector behavior. For instance, a commercial spam, a timely-seeming fishy email, a typical meeting email, and a project-status email. The output labels are represented as "Spam" and "Legitimate/Non-Spam" to stay consistent with the new frontend. The examples also demonstrate that, either high spam signs or a consistent combination of benign sender, phrasing and structural characteristics can lead to high confidence.

“TABLE V. REPRESENTATIVE EMAIL PREDICTION EXAMPLES”

Example	Subject / Pattern	Confidence	Output
E1	URGENT: Claim Your Cash Reward	98.6%	Spam
E2	Account action required within 30 minutes	97.4%	Spam
E3	Project Meeting Agenda - Friday	99.2%	Legitimate /

Example	Subject / Pattern	Confidence	Output
			Non-Spam
E4	Weekly project status and deliverables	98.9%	Legitimate / Non-Spam

6) Publication and Validation Path

The entire workflow should be included in this academic reporting, including the included public corpora, preprocessing rules, temporal split, hashing dimension, n-gram range, base-learner hyperparameters, DWM update rule, ADWIN confidence parameter, random seeds, and hardware environment. The time experiment could be repeated on various seeds and corpora to get more solid evidence of the generalisation of the adaptive ensemble across different benchmarks. The confusion-matrix counts must also be kept to allow independent checking of false positive behaviour on real messages.

The proposed technique needs to be evaluated with a common dataset and split against the exact MENACOMM-2025 SVC-MLP-Extra Tree ensemble. Such an experiment would distinguish between the effect of dynamic weighting based on drift and the effect of changes in the makeup of the corpus. In addition to the accuracy level, the comparison should also present information about false-positive rate, post-drift recovery time, inference latency, memory usage, and the stability of the results in case of a change in the campaign. A seemingly small increase in false positives may mean that valuable e-mail gets lost in the trash folder for the practical application of e-mail filtering.

V. CONCLUSION

In this research, we proposed a different framework of ensemble learning for email spam detection, which is different from the IEEE MENACOMM-2025 SVC-MLP-Extra Tree ensemble. It is suggested to use a combination of character n-gram feature hashing and Ridge, SGD-Huber and Passive-Aggressive base learners and Dynamic Weighted Majority and ADWIN drift detection. The architecture focuses on its adaptability to online interactions, minimal memory representation, false positive control and web based inference through Django. The new email-client UI displays the original message in a central reading window; the AI security analysis panel tags the message as Spam or Legitimate/Non-Spam, provides a score of confidence in the decision, and shows the most significant detection indicators. The comparative results indicate good accuracy, precision, recall and F1-score and the temporal evaluation the recovery time after a distribution shift is quicker. The method therefore provides a good basis for adapting email filtering when spam campaigns and spam terms are constantly changing.

REFERENCES

- [1] I. El Bitar, N. Abbas, M. Raad, F. Shmouri, and F. El Khechen, “Ensemble Learning-Based Approach for Email Spam Detection,” in Proc. 5th IEEE Middle East and North Africa Communications Conf. (MENACOMM), Byblos,

RESEARCH PAPER

- Lebanon, 2025, pp. 1–6, doi: 10.1109/MENACOMM62946.2025.10911030.
- [2] E. H. Tusher, M. A. Ismail, M. A. Rahman, A. H. Alenezi, and M. Uddin, “Email Spam: A Comprehensive Review of Optimize Detection Methods, Challenges, and Open Research Problems,” *IEEE Access*, vol. 12, pp. 143627–143657, 2024, doi: 10.1109/ACCESS.2024.3467996.
- [3] F. Jáñez-Martino, R. Alaiz-Rodríguez, V. González-Castro, E. Fidalgo, and E. Alegre, “A review of spam email detection: analysis of spammer strategies and the dataset shift problem,” *Artificial Intelligence Review*, vol. 56, pp. 1145–1173, 2023, doi: 10.1007/s10462-022-10195-4.
- [4] P. Charanarur, H. Jain, G. S. Rao, D. Samanta, S. S. Sengar, and C. T. Hewage, “Machine-Learning-Based Spam Mail Detector,” *SN Computer Science*, vol. 4, art. 858, 2023, doi: 10.1007/s42979-023-02330-x.
- [5] G. Nasreen, M. M. Khan, M. Younus, B. Zafar, and M. K. Hanif, “Email spam detection by deep learning models using novel feature selection technique and BERT,” *Egyptian Informatics Journal*, vol. 26, art. 100473, 2024, doi: 10.1016/j.eij.2024.100473.
- [6] K. I. Roumeliotis, N. D. Tselikas, and D. K. Nasiopoulos, “Next-Generation Spam Filtering: Comparative Fine-Tuning of LLMs, NLPs, and CNN Models for Email Spam Classification,” *Electronics*, vol. 13, no. 11, art. 2034, 2024, doi: 10.3390/electronics13112034.
- [7] F. E. Ayo, J. B. Awotunde, F. A. Kasali, et al., “A hybrid correlation-based deep learning model for email spam classification using fuzzy inference system,” *Decision Analytics Journal*, vol. 10, art. 100390, 2024, doi: 10.1016/j.dajour.2023.100390.
- [8] “Analysis of Spam Detection Techniques using Machine Learning,” in *Proc. 2024 Asia Pacific Conf. on Innovation in Technology (APCIT)*, 2024, doi: 10.1109/APCIT62007.2024.10673701.
- [9] R. Saravanan and V. Rathinasamy, “Email Spam Detection with Machine Learning and Vectorization,” in *Proc. 15th Int. Conf. on Computing Communication and Networking Technologies (ICCCNT)*, 2024, doi: 10.1109/ICCCNT61001.2024.10724545.
- [10] E. G. Dada, J. S. Bassi, H. Chiroma, S. M. Abdulhamid, A. O. Adetunmbi, and O. E. Ajibuwa, “Machine learning for email spam filtering: review, approaches and open research problems,” *Heliyon*, vol. 5, no. 6, e01802, 2019, doi: 10.1016/j.heliyon.2019.e01802.
- [11] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, “Online Passive-Aggressive Algorithms,” *Journal of Machine Learning Research*, vol. 7, pp. 551–585, 2006.
- [12] J. Z. Kolter and M. A. Maloof, “Dynamic Weighted Majority: An Ensemble Method for Drifting Concepts,” *Journal of Machine Learning Research*, vol. 8, pp. 2755–2790, 2007.
- [13] A. Bifet and R. Gavaldà, “Learning from Time-Changing Data with Adaptive Windowing,” in *Proc. SIAM Int. Conf. on Data Mining (SDM)*, 2007, pp. 443–448.
- [14] K. Weinberger, A. Dasgupta, J. Langford, A. Smola, and J. Attenberg, “Feature Hashing for Large Scale Multitask Learning,” in *Proc. 26th Int. Conf. on Machine Learning (ICML)*, 2009, pp. 1113–1120.
- [15] L. Bottou, “Large-Scale Machine Learning with Stochastic Gradient Descent,” in *Proc. COMPSTAT*, 2010, pp. 177–186.
- [16] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, “A survey on concept drift adaptation,” *ACM Computing Surveys*, vol. 46, no. 4, art. 44, 2014.
- [17] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, “Learning under Concept Drift: A Review,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 12, pp. 2346–2363, 2019.
- [18] B. Klimt and Y. Yang, “The Enron Corpus: A New Dataset for Email Classification Research,” in *Proc. European Conf. on Machine Learning (ECML) Workshop on Email Classification*, 2004.
- [19] V. Metsis, I. Androutsopoulos, and G. Paliouras, “Spam Filtering with Naive Bayes—Which Naive Bayes?” in *Proc. 3rd Conf. on Email and Anti-Spam (CEAS)*, 2006.
- [20] Apache Software Foundation, “SpamAssassin Public Corpus.” [Online]. Available: <https://spamassassin.apache.org/old/publiccorpus/>.
- [21] G. Forman, “An Extensive Empirical Study of Feature Selection Metrics for Text Classification,” *Journal of Machine Learning Research*, vol. 3, pp. 1289–1305, 2003.
- [22] F. Pedregosa et al., “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [23] T. Fawcett, “An introduction to ROC analysis,” *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.
- [24] T. Saito and M. Rehmsmeier, “The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets,” *PLoS ONE*, vol. 10, no. 3, e0118432, 2015.
- [25] B. W. Matthews, “Comparison of the predicted and observed secondary structure of T4 phage lysozyme,” *Biochimica et Biophysica Acta*, vol. 405, no. 2, pp. 442–451, 1975.
- [26] B. Krawczyk, L. L. Minku, J. Gama, J. Stefanowski, and M. Woźniak, “Ensemble learning for data stream analysis: A survey,” *Information Fusion*, vol. 37, pp. 132–156, 2017.
- [27] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, “MOA: Massive Online Analysis,” *Journal of Machine Learning Research*, vol. 11, pp. 1601–1604, 2010.
- [28] B. Biggio and F. Roli, “Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning,” *Pattern Recognition*, vol. 84, pp. 317–331, 2018.
- [29] M. Barreno, B. Nelson, A. D. Joseph, and J. D. Tygar, “The security of machine learning,” *Machine Learning*, vol. 81, pp. 121–148, 2010.
- [30] “Intelligent Email Spam Detection: A Machine Learning-Based Approach,” in *Proc. 5th Int. Conf. on Trends in Material Science and Inventive Materials (ICTMIM)*, 2025, doi: 10.1109/ICTMIM65579.2025.10988140.