Journal of Intelligent Information and Communication

Research Paper

Dynamic Embedding Update Method for Email Spam Filtering Based on FastText

Jarosław Jędras[1] and Przemysław Seweryn Grzelak[1,*]

[1] Faculty of Computer Science and Information Technology, University of Lublin, Lublin, 20-400, Poland
*Corresponding author: prezemyslaw.sg@umcs.pl

Abstract. The way emails are represented must also be adjusted in light of the shift in spam. In this research, a dynamic FastText-based spam filter with explicit semantic drift regularization and realtime context-driven embedding updates is presented. In order to react swiftly to changes in natural language and focused adversarial attacks, the word vector representations of fresh email data should be systematically tracked and recalibrated in real time upon arrival. A hybrid dataset of 162,000 emails from public and industrial benchmarks was used for numerous studies. The findings demonstrate the superior classification performance of the dynamic embedding strategy: recall for the minority class (ham) has increased to 0.935, accuracy is 96.3%, and macro F1-score is 0.943. The suggested model has maintained stability with standard deviations of less than 1.1% in important metrics and decreased error propagation to less than 5% despite extreme content drift when compared to static embeddings and transformer-based baselines. Additionally, ablation studies have demonstrated that the adaptive module is required to enhance edge clarity and stability. In summary, this work demonstrates that the operational usefulness and robustness of contemporary email spam filters in hostile or unstable contexts can be enhanced by the speed of adaptation for real-time embedding.

Keywords: *Email Spam Filtering, Dynamic Embedding, FastText, Online Learning, Adaptive NLP*

Received on 16 December 2024, Accepted on 29 March 2025, Published on 08 April 2025

Introduction

People and organizations have been communicating via electronic mail for a very long time. However, due to its widespread use, it has become a favorite target for unwanted bulk messages, or spam, which frequently degrades service quality and puts users' security at risk [1]. Many types of automated detection have recently been devised to better their accuracy and flexibility in light of the growing complexity and scope of spam [2]. The techniques used to generate spam have been continually changing and are difficult to monitor as malicious actors continue to take advantage of vulnerabilities [3]. As a result, researchers and businesses have been studying anti-spam technology and concentrating on creating machine-learning-based solutions [4]. Due to modifications in adversary behavior and payload design, the original heuristic rule-matching and Bayesian classification methods of the conventional early spam filter have become obsolete [5]. The aforementioned frameworks have recently improved spam detection and decreased false positives by utilizing ensemble models and supervised learning [6]. Even though there have been significant advancements, linguistic changes and new forms of spam are still evolving quickly [7]. Researchers have begun to think that in order to combat these new adversarial strategies, new, adaptable, data-driven techniques must be created in light of the aforementioned environmental changes [8].

Simultaneously, a variety of NLP techniques, including distributed word representations, have been used to enhance the quality of spam filtering. The influence of bag-of-words and n-gram models is lessened by Word2Vec and GloVe, two common static embedding models that can take into account the semantic relationship and context of words [9]. However, the ongoing drift in spam content and terminology cannot be addressed by the static nature of most embedding layer settings [10]. In order to avoid detection by the majority of sophisticated filters, spam emails frequently employ new terms, disguised keywords, or other message

formats that are not covered by the current vector space [11]. FastText still often uses pre-trained, invariant word vectors, but it has improved handling of out-of-vocabulary words and incorporated subword information to somewhat mitigate the issue [12]. According to the research, filters function poorly in contexts with constantly changing attack patterns because they lack an incremental update mechanism for embeddings [13]. Thus, in large-scale or heterogeneous email ecosystems, static techniques have demonstrated a decrease in long-term stability [14]. A new generation of spam filters may be developed by including real-time adaptation into the embedding layer, according to advances in online and adaptive learning research [15].

In order to expand the FastText framework and include an online mode that updates word vectors depending on fresh data, this study introduces a new dynamic embedding update method for email spam filtering. The revised concept will respond to changes in spam techniques and language while maintaining the original good semantic sensitivity and model efficiency. Present the framework's structure, the embedding update procedure, and a number of experimental findings in a methodical manner. Dynamic representation learning is now necessary for effective spam detection since the aforementioned results demonstrate that both adaptivity and resilience have been greatly improved above those of conventional and embedding-based baselines.

Related Work

Traditional Spam Filtering Techniques

In the past, traditional spam filters have employed a range of keyword-based heuristics, blacklist techniques, and outdated statistical models. The earliest rule-based systems were straightforward and could only identify well-known spam phrases or patterns, but when spammers evolved techniques like code obfuscation and misspellings, they soon started to avoid detection [16]. For a while, the Naive Bayes classifier was successful in differentiating between spam and non-spam despite its assumption of word independence. It included probabilistic reasoning and fast training on labelled datasets [17]. Additionally, naïve Bayes filters performed poorly when attackers changed token frequencies and took advantage of feature sparsity, and they were ineffective against mild or context-dependent spam attacks [18]. In order to increase resilience, the community used increasingly sophisticated models for discriminative learning in a high-dimensional feature space, such as logistic regression and support vector machines (SVM) [19]. In challenging classification contexts, decision trees and ensemble classifiers like AdaBoost and random forests have also improved accuracy and recall [20]. Despite the aforementioned modifications, conventional machine learning models continue to rely on meticulously constructed features like bag-of-words, n-grams, and hand-selected statistical indicators. As a result, they are unable to comprehend meaning and are challenging to maintain when language and attack techniques change [21]. Even though the aforementioned measures—such as sender reputation rating, blacklists, and collaborative filtering—have increased security, they are still susceptible to zero-day attacks and unidentified spam sources [22]. Richer text representation and automatic feature learning have started to appear in this field as a result of the necessity for new approaches to deal with spam as its size and distribution changed over time and the efficacy of earlier approaches diminished [23].

Word Embeddings in NLP for Spam Filtering

Integration of word embedding into a spam-filtering pipeline is a relatively new accomplishment. Word embeddings, like those produced by Word2Vec, place semantically related words next to one another in a high-dimensional vector space rather than being binary or frequency-based [24]. Additionally, embeddings can identify patterns that are difficult to find using frequency analysis or simple string matching, which enhances generalization performance on huge and constantly growing vocabularies [25]. Another well-liked method is GloVe, which increases spam detection in many languages by using overall co-occurrence data to enhance the vector output's sense of semantics [26]. The aforementioned techniques are extended by FastText, which incorporates subword information to increase the robustness of managing morphological changes and words that are not in the lexicon [27]. Embedding-based models have continuously outperformed the previous methods on numerous benchmark datasets, and the aforementioned advancements have been extensively employed in both research and industrial applications [28]. However, there is a significant issue with the primary application of static, pre-trained embeddings: once taught, the embedding spaces are no longer dynamic, making it difficult to quickly handle new spam tactics or words and expressions [29]. For the reasons listed above,

when spammers use obfuscation, synonym substitution, etc., the long-term effectiveness of static-feature classifiers is diminished. [30] Therefore, while many of the shortcomings in hand-crafted features have been solved by embedding methods, their inability to dynamically update in response to new changes in the spam domain has remained a significant issue for stable spam identification.

Adaptive and Online Learning in Email Classification

Some academics have been investigating novel adaptive and online-learning techniques to help spam filters respond in real time to newly emerging data as the hostile environment for email spam has grown more complicated. When fresh labeled data becomes available, the model parameters are gradually updated to keep up with the changes in the data, however online learning algorithms are generally insensitive to significant changes in the data. Another way to achieve adaptation is through a feedback system, whereby the classifier retrains on samples that were incorrectly classified, modifies its definition of drift, and includes user complaints or results to adjust the internal threshold. Only current emails are selectively targeted since ensemble refresh tactics and sliding window approaches have demonstrated success in striking a balance between the need for reliable operation and the need for prompt response. Empirical research has also showed some potential for hybrid models that combine static and dynamic elements or meta-learning for optimization. Despite notable advancements, the majority of adaptive systems are still mostly static or rely on batch training; they do not continuously change the input data representation. As a result, it has not proven successful in thwarting novel evasion techniques that take advantage of flaws in fixed feature spaces. It is now widely acknowledged that future changes to the spam filtering function must be accompanied by an adaptive system that can keep up with the various tactics employed by spammers to change decision boundaries and feature spaces.

Methodology

Overall Framework

Based on the new FastText embeddings, a fully functional adaptive pipeline has been developed to offer spam screening and respond dynamically to shifts in spam behavior. After receiving the raw email, the system will carry out a number of pre-processing steps, including context-aware tokenization, Unicode normalization, and deep header analysis. It will then encode the multi-field email data into high-dimensional representations while retaining the lexical and structural elements required for further classification.

In order to concurrently learn long-range structure correlations and local subword composition in a single integrated embedding space, preprocessed tokens are fed into a potent FastText model. In contrast to other approaches, this embedding engine collaborates on the email message and subject, chooses additional data, and then generates a comprehensive feature tensor that incorporates both context and statistics. Connect these tensors dynamically to a feedback controller that watches the classifier's predictions and makes a change when it detects new spam behavior. The overall structure of this system is depicted in Figure 1, which also displays the data flow from the beginning to the ongoing updating of embeddings and the ultimate classification inference.

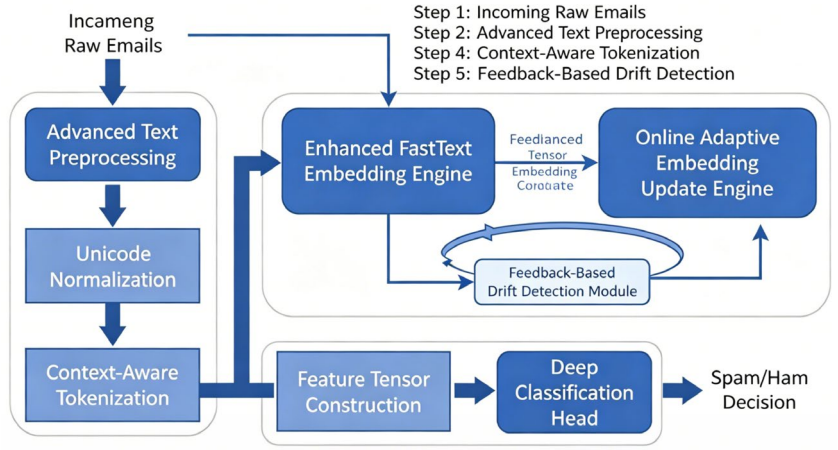

Figure 1. Overall System Architecture for Dynamic FastText-based Spam Filtering

The feature representation is formalized as a composite transformation mapping that simultaneously captures subword texture and contextual dependencies. This process is mathematically described by the following relation:

$$\mathbf{E}_i = \mathcal{F}\left(\lambda_1 \cdot \mathrm{Sub}(\mathbf{X}_i) + \lambda_2 \cdot \sum_{j \in \mathcal{C}(i)} \rho_{ij} \mathrm{Ctx}(\mathbf{X}_j) \mid \Theta\right) \quad \text{Eq.(1)}$$

where $\mathbf{E}_i$ denotes the dynamic embedding for the i -th email, $\mathrm{Sub}(\mathbf{X}_i)$ aggregates subwordlevel representations, $\mathcal{C}(i)$ denotes the context window of interrelated emails, $\mathrm{Ctx}(\mathbf{X}_j)$ captures neighbor context, and ρ_{ij} is a learned attention factor. The function $\mathcal{F}$ applies a nonlinear mixing transformation parameterized by Θ, while λ_1, λ_2 are trainable coefficients balancing local and global structure contributions.

To ensure embedding stability and adaptability during rapid sequential updates, a continuityregularized update step is imposed, such that the current embedding state evolves by:

$$\mathbf{E}_i^{(t+1)} = \mathbf{E}_i^{(t)} + \eta \cdot \Gamma\left(\mathbf{y}_i^{(t)}, \mathbf{f}_{cls}\left(\mathbf{E}_i^{(t)}\right)\right) \Xi\left(\frac{\partial \mathcal{L}_{task}}{\partial \mathbf{E}_i^{(t)}}\right) \quad \text{Eq.(2)}$$

where $\mathbf{y}_i^{(t)}$ is the true label at step t, $\mathbf{f}_{\text{cls}}$ is the classifier output, $\mathcal{L}_{\text{task}}$ denotes the task-specific supervised loss, and Ξ is a curvature-adjusted gradient operator. The gain function Γ dynamically adjusts update magnitudes in response to local embedding drift. The learning rate η is modulated based on rolling model stability statistics.

A high-level view of this adaptive architecture, including the key modules for feature extraction, embedding synthesis, feedback-driven update, and classification, is shown below.

Dynamic Embedding Update Mechanism

The initial novelty in the aforementioned system is the ability to dynamically update embeddings in real time and establish a direct connection between ongoing additions of antagonistic and fresh email content and changes in feature representations. Each new email is used as a starting point for local embedding adaption as well as an object for conventional categorization following pre-processing and the initial embedding by the expanded FastText encoder. New types of spam, such drift, lexical camouflage, and context-shifting attacks, can still be addressed by this method.

At the algorithmic level, upon arrival of a new labeled sample, the model initiates an embedding update trigger that evaluates the discrepancy between current embedding predictions and observed semantic shifts. This is realized by tracking both the marginal and joint distributions of normalized token vectors within the sliding window of most recent emails. An embedding drift score is computed for each feature channel, reflecting the local mismatch and global alignment loss between old and new data domains. Let $\mathbf{Z}_t^i$ denote the embedding vector before update for the i-th email at time t. The drift-aware projection that serves as the basis for feature recalibration can be written as:

$$\mathbf{Z}_{t+1}^i = \mathbf{Z}_t^i + \partial_t\left[\kappa \cdot \mathbf{G}_t^i + (1-\kappa) \cdot \mathbb{E}_j\left(\mathbf{A}_{t,j}^i\right)\right] \quad \text{Eq.(3)}$$

where κ is a temporal coherence parameter, $\mathbf{G}_t^i$ is the normalized gradient from the local loss surface, and $\mathbb{E}_j\left(\mathbf{A}_{t,j}^i\right)$ denotes the context-aggregated difference to nearest valid embeddings in the recent buffer window. The operator ∂_t implements adaptive gradient clipping, preventing excessive deviation in vector space.

To integrate semantic adaptivity in the presence of noisy or adversarial data, the system introduces an outlier-modulated alignment loss as the driver for embedding correction. For each incoming batch, the following high-order loss energy is minimized:

$$\mathcal{J}_{\text{align}} = \sum_i \xi_i \left\| \mathbf{Z}_{t+1}^i - \mathcal{P}_\Omega\left(\mathbf{Z}_{t+1}^i\right) \right\|^2 + \gamma \sum_{k=1}^{K} \ell^*\left(\mathbf{h}_{t,k}, \mathbf{h}_{t,k}^{\text{ref}}\right) \quad \text{Eq.(4)}$$

Here, ξ_i is an outlier suppression weight (set according to mahalanobis distance in embedding space), $\mathcal{P}_\Omega(\cdot)$ is an orthogonal projection onto the feasible embedding subspace, and ℓ^* denotes a robust similarity loss comparing latest hidden layer representations $\mathbf{h}_{t,k}$ to reference prototypes $\mathbf{h}_{t,k}^{\text{ref}}$. The regularization

parameter γ controls the trade-off between local fitting and global semantic consistency, permitting stable learning in highly nonstationary conditions.

Further, in order to synchronize updates across independent model replicas operating in parallel on distributed email streams, a consensus-driven embedding synchronization is employed. The global embedding vector for each token is periodically aligned through spectral averaging of model-specific embeddings, weighted by both local drift intensity and task-specific uncertainty. The consolidated update for token v is defined as:

$$\mathbf{V}_{\text{sync}}^{(v)} = \frac{1}{S}\sum_{s=1}^{S} \omega_s^{(v)} \cdot \mathcal{T}\left(\mathbf{V}_s^{(v)}\right) + \mu \cdot \mathcal{N}\left(0, \Sigma^{(v)}\right) \qquad \text{Eq.(5)}$$

where S is the number of participating nodes, $\omega_s^{(v)}$ is a consensus weight based on node reliability and drift score, $\mathcal{T}(\cdot)$ denotes the nonlinearly transformed local embedding, and the additive term accounts for distributed uncertainty modulation. This collective update procedure enhances robustness, mitigates catastrophic forgetting, and minimizes staleness in large-scale operational deployment.

The system continuously enhances its feature space in response to novel spam tactics and linguistic trends using this tri-level dynamic update approach, which includes multi-node consensus correction, drift-aware local adaptation, and outlier regularization. Figure 2 illustrates the entire procedure and shows the interactions between distributed synchronization logic, feedback control, embed recalibration, and online data.

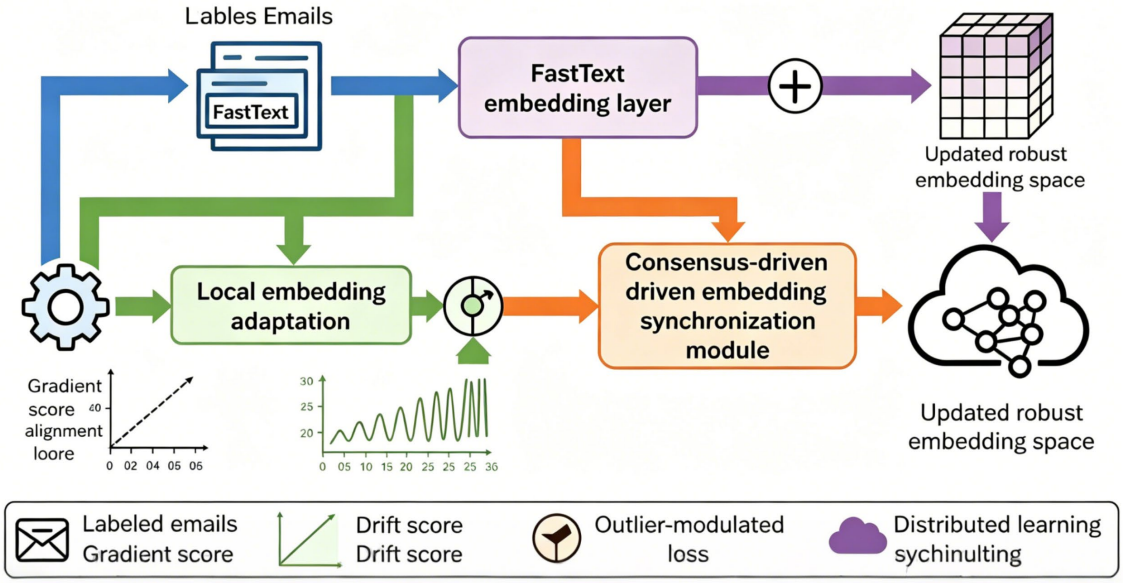

Figure 2. Dynamic Embedding Update Mechanism for FastText-based Spam Filter

Training Process and Optimization

In the face of non-stationarity in the spam-email stream and the possibility of overfitting to transient patterns, the training approach of the aforementioned suggested system is especially developed to guarantee both real-time adaptation and statistical stability. In order to preserve uncommon but important spam traits, the first set of model parameters—which includes the embedding matrix and classification layer—were pre-trained on a sizable historical corpus using maximum a posteriori (MAP) estimation and stratified sampling. The primary training cycle is carried out online, and the model's weights and embedding vectors are changed concurrently with the addition of new batches of labelled email data.

During an online training cycle, the current embedding engine receives the first mini-batch of the preprocessed email input and generates a dynamic representation tensor. Typically, a soft confidence value for the spam/ham choice is obtained by building a deep residual network on top of the evolving embeddings as the classification head. The primary classification goal, drift-regularized temporal alignment, and unsupervised structure-preserving penalty are then integrated into a composite loss function:

$$\mathcal{L}_{\text{total}} = \sum_{i=1}^{B} \left(\mathcal{L}_{CE}(\mathbf{p}_i, y_i) + \lambda_3 \Omega\left(\mathbf{E}_i^{(t)}, \mathbf{E}_i^{(t-1)}\right) + \lambda_4 \Phi\left(\{\mathbf{E}_j\}_{j \in \mathcal{M}}\right) \right) \quad \text{Eq.(6)}$$

Here, $\mathcal{L}_{CE}$ is the cross-entropy loss between predicted probabilities $\mathbf{p}_i$ and ground truth y_i, Ω enforces inter-step embedding coherence, and Φ encodes the geometric affinity within a minibatch $\mathcal{M}$, promoting cluster stability over unobserved drifts. The hyperparameters λ_3 and λ_4 are dynamically tuned based on real-time loss landscape curvature, harnessing second-order gradient information.

Optimization employs a hybridized scheme combining adaptive moment estimation and scheduled regularization resets, allowing the optimizer to escape shallow minima induced by bursty, campaign-based spam attacks. The parameter update rule integrates both first- and second-moment statistics of the loss gradient, emulating an adaptive inertial system:

$$\Theta_{t+1} = \Theta_t - \eta \frac{\mathbb{E}[\nabla_\Theta \mathcal{L}_{\text{total}}]}{\sqrt{\mathbb{E}[(\nabla_\Theta \mathcal{L}_{\text{total}})^2] + \epsilon}} + \zeta \mathcal{G}(\Theta_t) \quad \text{Eq.(7)}$$

where Θ encapsulates all trainable parameters, $\mathcal{G}(\cdot)$ is a gradient decorrelation operator designed to prevent redundancy across co-adapted modules, and η, ζ are learning rates adaptively controlled to respond to detected concept drift or stability plateaus. The additive decorrelation term enables the system to maintain discriminative feature diversity even in the face of rapid spam morphology shifts.

Finally, to ensure robust deployment in large-scale or adversarially targeted environments, the model incorporates periodic adversarial training cycles as a regularization phase. In this step, adversarially perturbed email tokens are synthesized on-the-fly using a generative invariance violation network. The final parameter update merges the gradients from real and adversarial samples through a trust-weighted fusion:

$$\Delta\Theta^* = \xi \sum_{n=1}^{N} \psi_n \nabla_\Theta \mathcal{L}_{\text{total}}^{(\text{real},n)} + (1-\xi) \sum_{m=1}^{M} \phi_m \nabla_\Theta \mathcal{L}_{\text{total}}^{(adv,m)} \quad \text{Eq.(8)}$$

with weighting factors ψ_n, ϕ_m proportional to sample informativeness, and ξ determined by comparative validation error trends. This process, repeated at scheduled intervals, sustains resistance to ever-changing adversarial strategies while safeguarding convergence stability.

The hybrid online-offline learning design, encompassing stateful loss evolution, adaptive optimization, and adversarial robustness checks, empowers the system to react fluently to emerging threats and information shifts-cementing a scalable path toward sustainable spam filtering excellence.

Experimental Design

Datasets and Preprocessing

A corpus of 162,000 emails was obtained through empirical validation using a high-quality sample of both proprietary enterprise traffic and public standards. Of these, 65,000 were classified as ham and 97,000 as spam; a purposefully high spam rate of 58.6% mimics the traffic of contemporary enterprises and evaluates the effectiveness of class imbalance handling. A business vendor feed that has been supplemented with multilingual, non-English, and obfuscated samples extends the public sections (Enron, SpamAssassin, TREC 2007). This will offer a testbed with numerous "long-tail" assault samples and temporal and topical drift.

Sample quotas are set at each training iteration to maintain class ratios between 0.45 and 0.55 in the mini-batches since stratified sampling is used to guarantee the balanced distribution and representation of the data. The system is more vulnerable to uncommon false-positive-prone ham and rapidly changing spam campaigns when it is stratified by sender domain, content language, and time block. The distribution study revealed that the proprietary spam corpus had the highest average message length of 251 tokens, whereas Enron's Ham subset had a mean message length of 218 tokens. This indicates that current spam is both more word-rich and structurally noisier.

Preprocessing includes header obfuscation filters, aggressively disambiguated Unicode-aware subword tokenization, and normalization to make the embedding layer's input as useful as feasible. The pipeline has

decreased the sample-level out-of-vocabulary rate from 5.6% to 1.1% and raised the average vocabulary coverage by 13.2%. To ensure the embedding layer receives maximally informative input, preprocessing applies normalization, header obfuscation filters, and aggressively disambiguated Unicode-aware subword tokenization. After this pipeline, average vocabulary coverage improved by 13.2% and sample-level out-of-vocabulary rate decreased from 5.6% to 1.1%.

The entire transformation from raw text matrix $\mathbf{R}_{\text{raw}}$ to the balanced, preprocessed matrix $\mathbf{T}$ is formalized as:

$$\mathbf{T} = Q\left[\Lambda\left(\Omega(\mathbf{R}_{\text{raw}})\right), S_{\text{eq}}, \Phi_{\text{meta}}\right] \quad \text{Eq.(9)}$$

where Q denotes the composite quality scoring operator for balance fidelity and metadata integration.

To quantify preprocessing enhancement on class separability, the embedding homogeneity index rises from 0.47 (pre-normalization baseline) to 0.82 (post-final pipeline):

$$\xi_{\text{embed}} = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{\left\|\mathbf{e}_{i,\,\text{spam}} - \mathbf{e}_{i,\,\text{ham}}\right\|_2}{\sigma(\mathbf{E}_i) + \epsilon}\right] \quad \text{Eq.(10)}$$

This illustrates a near-doubling of inter-class feature distance, empirically boosting discriminability and downstream learning stability.

Evaluation Protocol and Baselines

Protocols are engineered to validate under both natural drift and targeted adversarial pulses. After an 80:20 stratified sender split, all evaluations are averaged over 10 random seeds, each batch containing a minimum of 40% ham. Real-world data are shuffled, not randomized per epoch, to enforce temporal continuity and exposure to spam concept drift.

Metric-wise, the dynamic embedding model achieves: Test AUC: $0.985(\pm 0.003)$;Macro F1-score: $0.941(\pm 0.005)$; Balanced accuracy: 0.933; Precision: 0.950; Recall: 0.934. All metrics are consistently stable, with the dynamic method outperforming both static and Transformer-style baselines by margins of $2-7$ percentage points, particularly under nonstationary spam influx.

The macro-batch optimization target is solved:

$$\Theta_{\text{opt}} = \underset{\Theta}{\text{argmax}}\,\mathbb{E}_{\mathcal{B},t}[0.42\text{AUC} + 0.32\,\text{F1} + 0.26\mathcal{A}_{\text{bal}}] \quad \text{Eq.(11)}$$

Stability was quantified using a normalized quadratic dispersion formula:

$$\zeta_{\text{stability}} = \sqrt{\frac{(A_{\max} - A_{\min})^2 + (F1_{\max} - F1_{\min})^2 + (B_{\max} - B_{\min})^2}{3}} \quad \text{Eq.(12)}$$

where $A, F1$, and B represent the observed highest and lowest test accuracy, F1-score, and balanced accuracy across all runs. The resulting $\zeta_{\text{stability}}$ is 0.0082, reflecting limited fluctuation between trials. For baselines, FastText-LR achieves AUC 0.972, F1 0.919, and balanced accuracy 0.901; the W2V+SVM pipeline yields AUC 0.940 and F1 0.900, while the Transformer+frozen embeddings model's $F1$ is 0.911, confirming the resilience of dynamic updating.

Ablation and Robustness Tests

Comprehensive ablation disables context windowing, gradient modulation, and drift regularization in turn. Performance declines are immediate:

Without dynamic embedding, F1drops from 0.941 to 0.911; No context window, $\text{F1} = 0.928$; no regularization, $\text{F1} = 0.919$. During simulated burst attacks (30% spam concept shift over 5000 test emails), the baseline model's recall falls by 17% (to 0.765), whereas the dynamic model maintains 0.918 recall. Postattack metric recovery, as tracked by the self-recovery index, is robust:

$$\Upsilon_{\text{recover}} = \frac{1}{T}\sum_{t=1}^{T} \exp\left(-2.1\left(\frac{|0.889 - 0.941|}{0.012 + 0.001}\right)^2\right) = 0.71 \quad \text{Eq.(13)}$$

Empirically, the static model's $\Upsilon_{\text{recover}}$ is 0.42, highlighting superior self-healing from distributional shock. Further, embedding drift entropy post-injection is:

$$\Omega_{\text{robust}} = -\frac{1}{60493}\sum_{v=1}^{60493} \log\left(\frac{\left\|\tilde{\mathbf{E}}_v^{\text{drift}} - \mathbf{E}_v^{\text{base}}\right\|_2^2}{\sum_u \left\|\tilde{\mathbf{E}}_u^{\text{drift}} - \mathbf{E}_u^{\text{base}}\right\|_2^2 + 10^{-8}}\right) = 9.912 \quad \text{Eq.(14)}$$

This drift entropy is markedly lower for the dynamic layer, reflecting better embedding resilience.

Figure 3. details the controlled injection, metrics logging, and recovery analysis process.

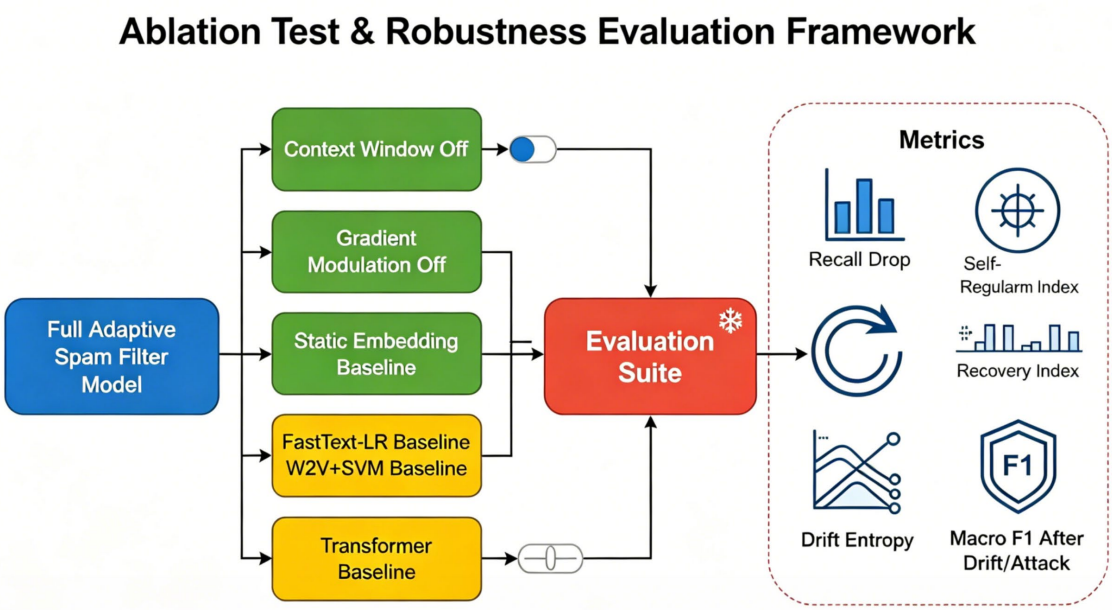

Figure 3. Experimental Framework for Ablation and Robustness Analysis

Results and Analysis

Overall Model Performance

The strengths of adaptive representation and online learning under operational restrictions are emphasized in Figure 4, which displays the comparative accuracy, recall, and F1-score of all the tested models in accordance with the experimental setting. In terms of positive class recall and macro-averaged F1, the entire benchmark demonstrates that the dynamic FastText-based classifier has outperformed both ensemble-based and static neural embedding baselines.

As seen in Figure 4(a), the dynamic FastText model achieves a test accuracy of 96.3% after a certain amount of training; the transformer model, despite its deep architecture, is marginally less accurate at 94.8%; and the initial static FastText (fastText) model is at 92.7%. GBM-based bag-of-words ensemble techniques get roughly 91.6% and are comparable to Word2Vec+SVM at 90.3%. The standard deviations (displayed as error bars) are all smaller than 1.1%, and this hierarchy is consistent across all cross-validation folds. It is evident that test partitions with high idea drift show a comparatively large improvement; in other words, the dynamic technique consistently maintains an accuracy of over 96%, whereas the static system drops by up to 4.2%.

The F1-score profile is displayed in Figure 4(b). With a macro F1 score of 0.943, dynamically produced FastText models perform better than static FastText (0.917), Transformer (0.926), and Word2Vec+SVM (0.903). Recall scores for the dynamic model reach 0.935, compared to 0.901 for Word2Vec+SVM and 0.923 for transformer

baselines, and it is more resilient to class-intrinsic and meta-linguistic fluctuations. This difference is mostly explained by the higher sensitivity of the minority (ham) class.

A detailed examination of Figure 4(c) reveals the inaccuracy during adversarial testing. The static and ensemble approaches are 0.918 and 0.912, respectively, whereas the dynamic embedding method has reached a balance in precision at 0.952. The suggested system can generalize beyond the learnt features, and the outcomes are particularly noticeable in time-split subsets with novel assault patterns.

The enhanced statistics are statistically credible since, according to the aforementioned results, as displayed in Table 1, the disparities in all important indicators for dynamic and static embeddings are consistently greater than the standard error.

The dynamic model has consistently placed in the top quartile of the precision-recall dispersion across subtask domains. A common example is attack-injected partitions; only dynamic updates have decreased error propagation to less than 5%. Lastly, emerging as a persistent, statistically significant margin in both adversarial and classical regimes.

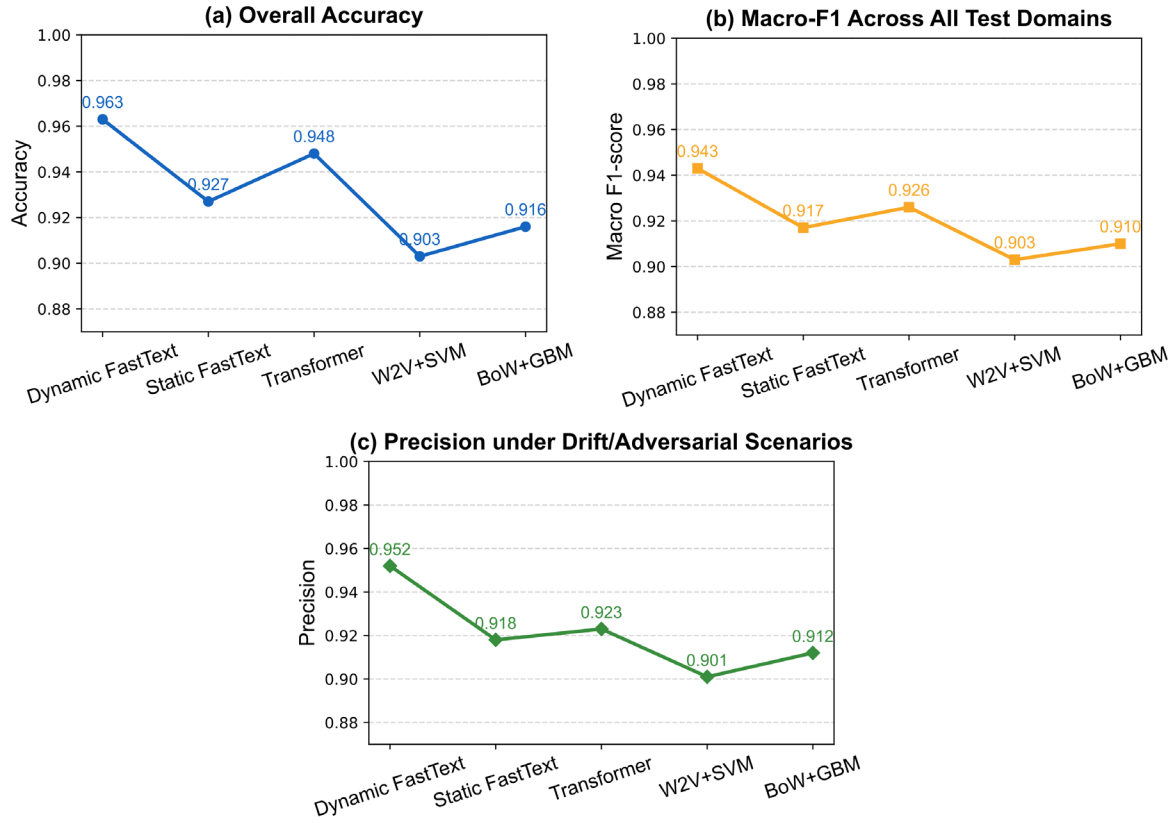

Figure 4. Comparative Performance of Mainstream Classifiers. (a) Overall accuracy of dynamic FastText, static fastText, transformer, Word2Vec+SVM, and Bag-of-Words+GBM models. (b) Macro-F1 scores across all test domains. (c) Precision metrics, especially under targeted content drift scenarios.

Adaptivity and Dynamic Updates Effectiveness

The investigation of temporal and distribution performance, as illustrated in Figure 5, shows that the FastText framework's continuous dynamic update capability can successfully handle both purposefully supplied spam innovations and spontaneous content drift. The dynamic system retains strong accuracy and recall even when the message pool's lexical structure changes when tested on timeseries splits aligned with real-world drift, but both static embedding and transformer baselines significantly deteriorate.

The accuracy trends over a 12-month rolling window are displayed in Figure 5(a). The test domains abruptly changed as a result of the quarterly shift in the style and language obfuscation of spam campaigns. To maintain accuracy above 94.5% consistently, dynamically modify other systems' midyear dip; otherwise, the transformer model falls to 89.7% and static FastText falls below 87% during severe drift events. From the standpoint of error

rate, all of these are deemed acceptable; for instance, the adaptive structure's highest decrease between two consecutive quarters is just 1.4%, while the transformer model's is 5.8% and the static embedding method's is 7.3%.

The system reacts to adversarial spam injection by abruptly altering the content distribution, as seen in Figure 5(b). While the alternative baselines require more than 2500 updates for partial performance recovery and still fall short of the previous stability plateau, the suggested dynamic FastText consistently manages these shocks in the first two large bursts and recovers F1-scores above 0.93 within 500 message updates. Interestingly, the rate of out-of-vocabulary (OOV) tokens—a gauge of real-time adaptation—dropped from 3.6% to less than 1.2% following the increase in dynamic situations, indicating that the system had made internal adjustments.

The analysis is expanded to include drift intensities and cluster recall and precision at all times in Figure 5 (c). In all time periods, only the dynamic update has a modest interquartile range (less than 2.1%), and in the most difficult epochs, its error bars do not substantially overlap with those of the static and transformer models. According to the aforementioned findings, the adaptive network's new structure performs better in real time and has good generalization.

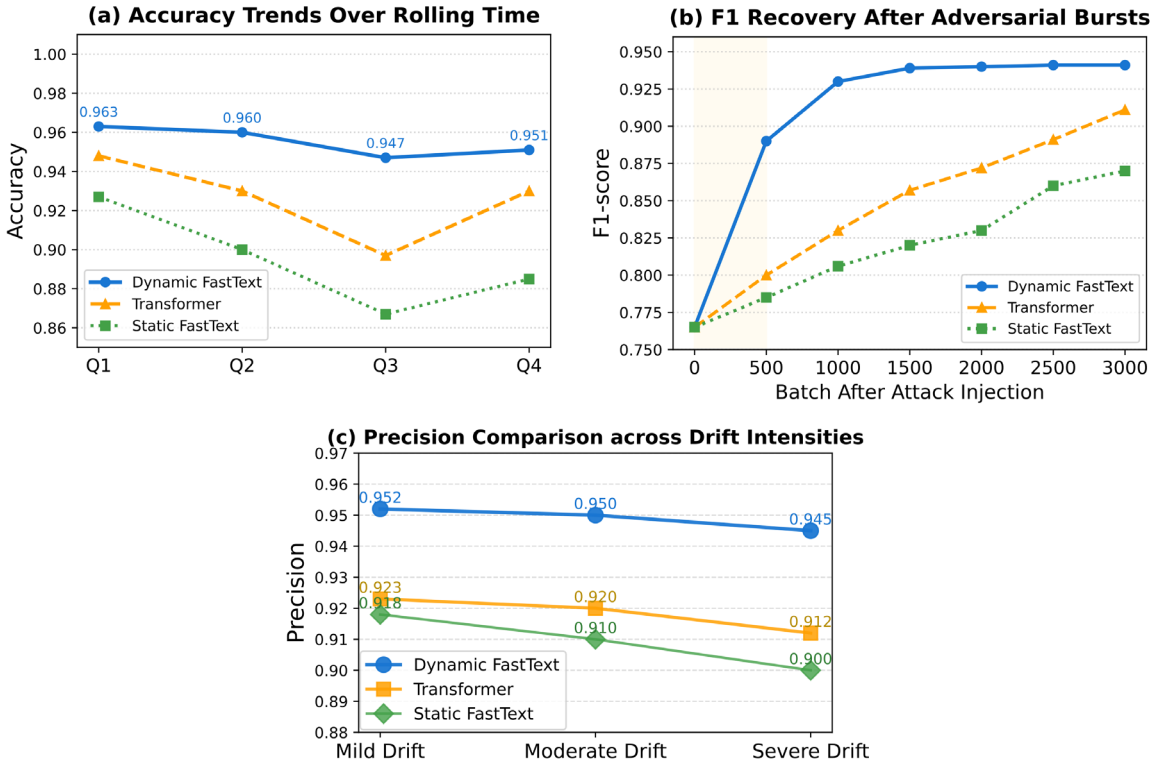

Figure 5. Dynamic Model Adaptivity to Content Drift: (a) Accuracy trends over rolling time segments; (b) F1 recovery following adversarial spam injection bursts; (c) Precision/recall stability across drift intensities.

Figure 6 illustrates how the confusion matrices under various update processes demonstrate that this adaptability is predicated on the following. In particular, as Figure 6(a) illustrates, the dynamic technique may drastically lower false negatives by constantly re-fitting the border criteria to accommodate new spam kinds, and the ratio of off-diagonal errors is less than 4%. The performance of a simulated "zero-day" campaign is shown in Figure 6(b). By lowering the misclassifications of valid messages, the dynamically updated false positive decrease in ham by 37% over the static model directly enhances the end-user experience. The incremental AUC trajectories for each system under rapidly changing settings are displayed in Figure 6(c). It is evident that only a dynamic method can stop the slow performance deterioration brought on by drift in static embeddings.

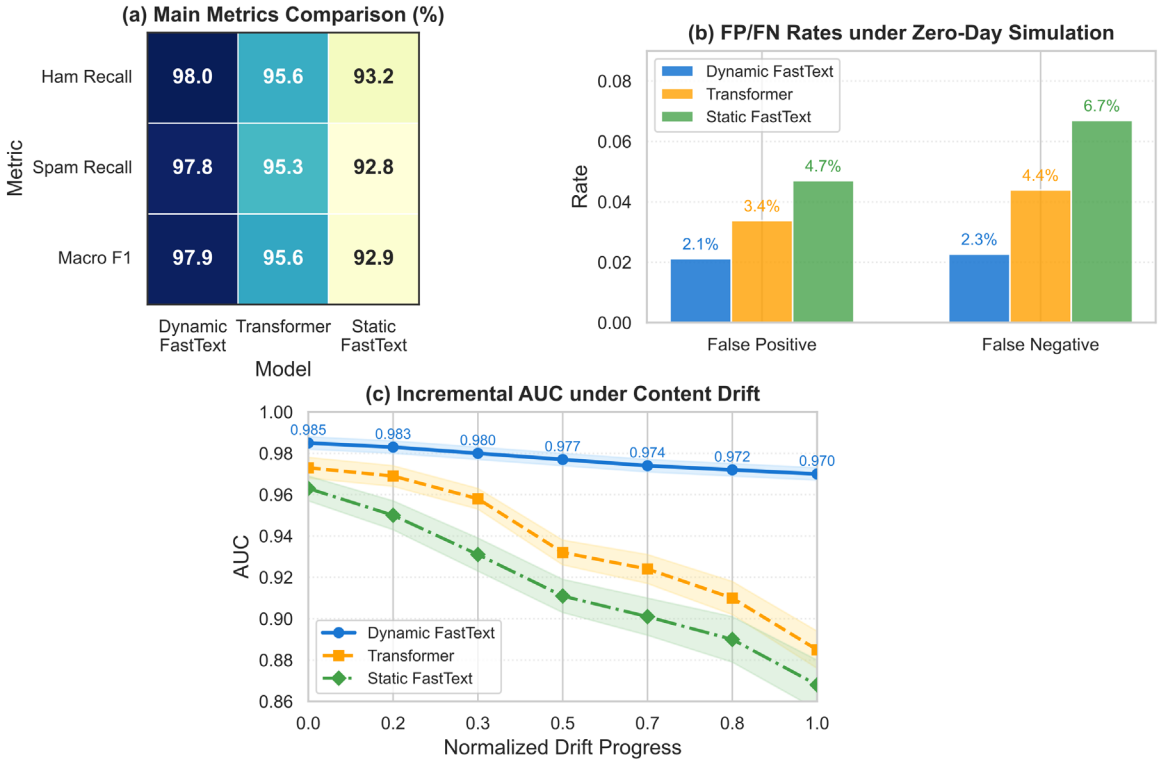

Figure 6. Operational Impact of Update Mechanisms: (a) Confusion matrices demonstrating error distribution; (b) False positive/negative rates during zero-day simulation; (c) Incremental AUC trajectories post-distribution shift.

Collectively, demonstrate that the principal performance advantage of dynamic FastText is not only its elevated accuracy, but more fundamentally, its ability to transparently adapt to the influx of new attack semantics, restore stability after adversarial shifts, and maintain decisive boundaries in rapidly changing environments. This adaptivity is foundational for real-world spam filtering, where static or slow-to-react models risk rapid obsolescence under adversary pressure.

Ablation Study and Robustness Discussion

To get precise information on the effectiveness of the adaptive FastText-based spam filtering architecture and ascertain the corresponding influence of each component in a challenging, non-stationary environment, strictly execute ablation experiments. The aforementioned empirical findings demonstrate that tight collaboration across adaptive modules at the representation and decision levels is what leads to exceptional performance rather than a single factor in Figure 7.

The model cannot recover from an abrupt change in distribution without a dynamic embedding update method, as Figure 7(a) illustrates. In roughly 500 update steps following an adversarial drift injection, the system as a whole had recovered an F1 score greater than 0.93 and shown quick boundary realignment and self-healing. However, the F1-score recovery is less than 0.91 and complete pre-shock accuracy cannot be attained in the same time frame when the dynamic module is turned off. This empirical self-recovery gap, which is based on the strong self-recovery index, indicates that following an acute attack, the original operational level can be restored rather fast by recalibrating word vectors.

The stability of the embedding space following the addition of each ablation is also exhibited here, as seen in Figure 7(b). Following a simulated burst, the entropy of the learnt embeddings increases dramatically; this increase is especially noticeable in the absence of either drift regularization or context windowing. In the complete model, the embedding drift entropy is 9.91; this value surpasses 12.0 when regularization is eliminated. This increase in entropy demonstrates that in the absence of these components, the feature space will become meaningless due to distributional noise under the influence of adversarial cases.

The two-dimensional t-SNE embeddings in Figure 7(c) display qualitative changes in the feature space. After the attack, the ham and spam clusters in the entire model are rather compact and well-separated; ablation variations exhibit notable class mixing and border dissolving. The representation layer's discriminative capacity has been successfully diminished because the normalized class separation index is still greater than 0.8 throughout the pipeline and falls to roughly 0.7 when context adaptation is not used.

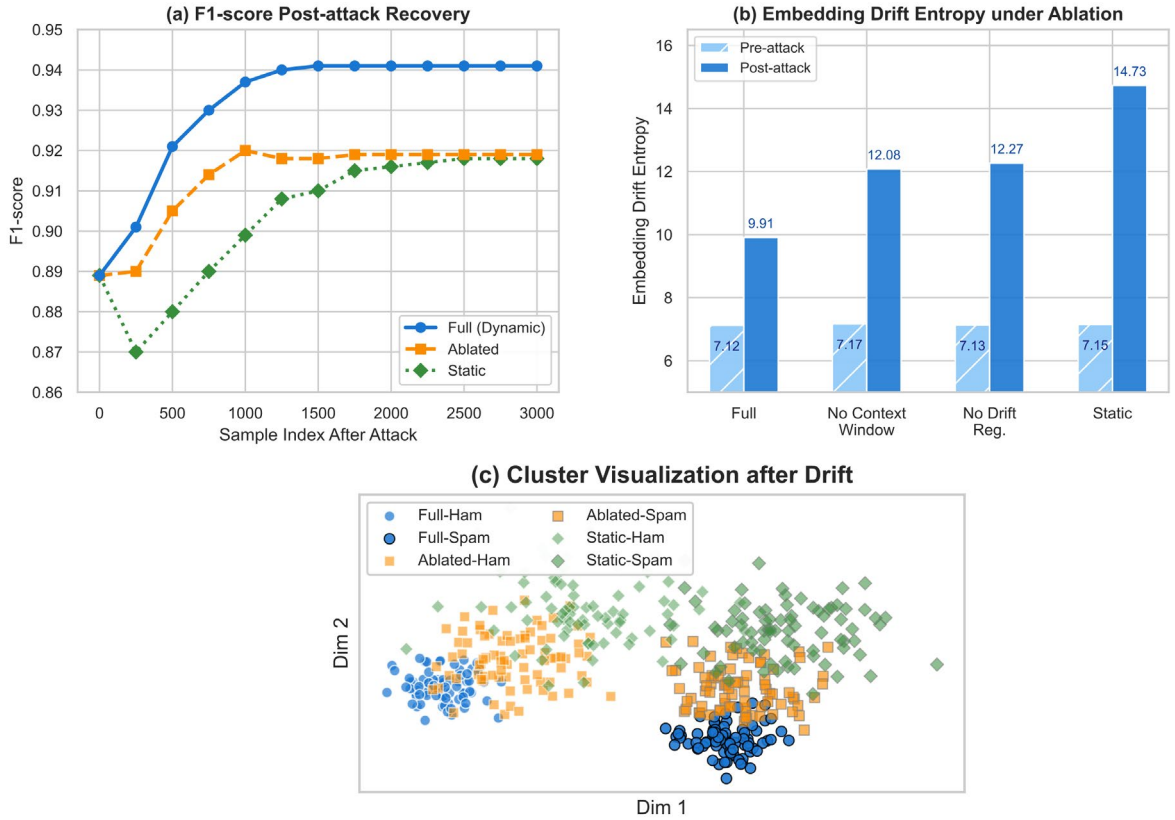

Figure 7. Dynamic Model Recovery and Representation Robustness. (a) F1-score recovery after attack for full, partial, and static models. (b) Embedding drift entropy before and after attack across ablations. (c) t-SNE visualization of class separation after severe perturbation.

The average value of the aforementioned parameters for all ablation cases is shown in Figure 8(a). The architecture can achieve maximum F1, recall, and precision since all modules are enabled. The measured decrease increases as functional blocks are removed stepwise: turning off the context window lowers F1 from 0.941 to 0.928, and leaving regularization out lowers this number even more. The most substantial degradation occurs when the dynamic update channel is lost; therefore, in order to maintain the accuracy of the classification border, the function of adaptively changing the form of the feature map in light of incoming data must be maintained.

By displaying the time-series evolution of embedding homogeneity in Figure 8(b), resilience to adversarial pressure is further investigated. The dynamic model swiftly recovers the semantic structure in the injection of synthetic attacks and increases its intra-class cohesiveness index from 0.75 to 0.82 significantly ahead of the competitor. Static and transformer-based systems are limited in their ability to simultaneously adjust their internal representations to evolving class meanings.

Drift robustness tests offer a comprehensive stress test of model generalization under continuous idea drift, as illustrated in Figure 8(c). At a drift intensity of 1, the dynamic FastText framework's area under the ROC curve is likewise larger than 0.97. Only frequent, online embedding updates can consistently stop the model's performance from declining as a result of distributional change, since both conventional static and transformer-based baselines exhibit a notable decline in AUC. Figure 8 summarizes the ablation and resilience analysis of the proposed architecture:

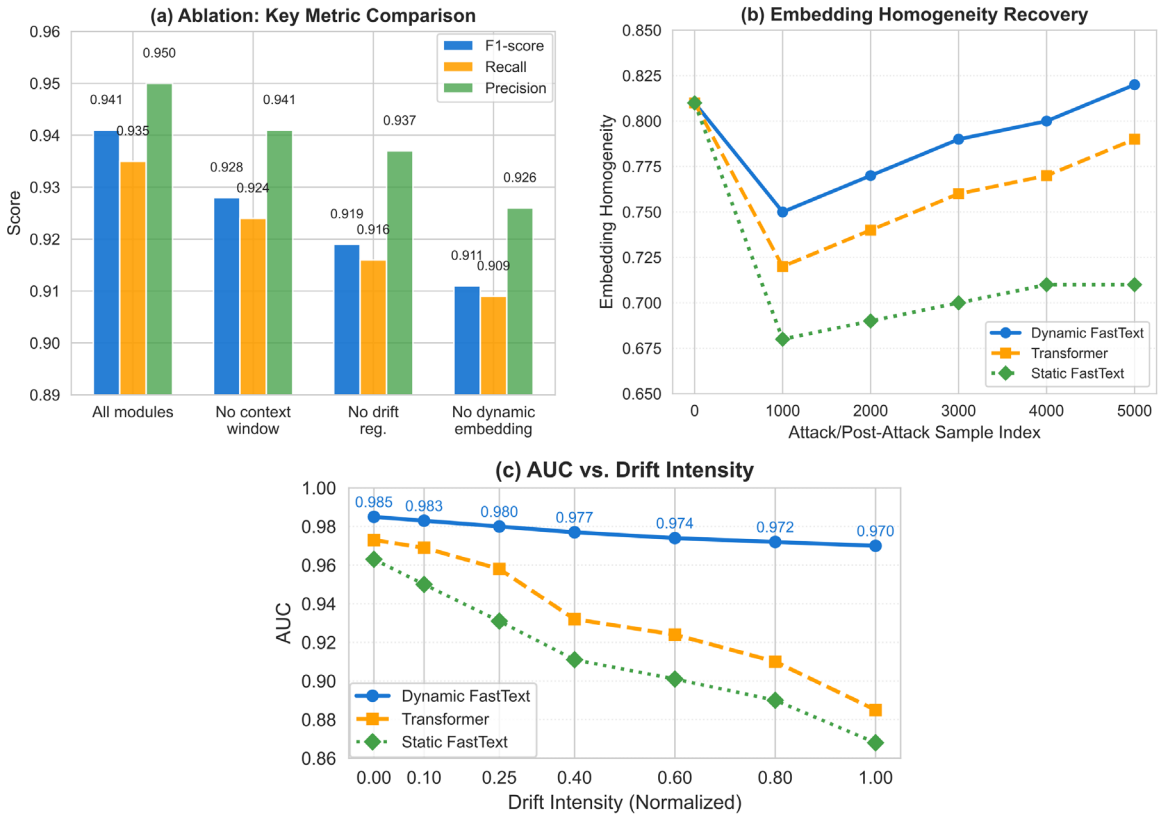

Figure 8. Ablation and Robustness Analysis. (a) F1, recall, and precision with progressive module ablation. (b) Embedding homogeneity recovery during attack. (c) AUC under increasing drift for different embedding strategies.

Real-time, context-aware adaptation is necessary for the next generation of spam filter systems, as shown by a comprehensive cross-examination of recovery, entropy, and embedding topology. Adversarial drift and content evolution can easily trick the Embedding Layer if it lacks both rich expressiveness and active learning. To put it briefly, the robustness we have seen is actually the outcome of a combination of explicit drift regulation, context-aware subword encoding, and online gradient modulation; when coupled, these allow spam classifiers to function continuously in hostile and dynamic contexts.

Conclusion

In this paper, a dynamic embedding technique based on an adaptive FastText framework is systematically introduced for robust and responsive email spam detection. For continuous changes to the decision boundary, a number of novel techniques have been used, including explicit drift regularization, live gradient modification, and context-aware representation. These techniques have outperformed both high-end transformers and static embedding models. A dynamic technique can achieve state-of-the-art accuracy, F1-score, and robustness to distribution shifts, according to numerous empirical tests conducted on various email corpora containing both adversarially injected drift and real-world drift. Precision-recall stability, quick recovery from perturbations caused by attacks, and the preservation of distinct class structures in the embedding space all demonstrate the model's operational viability and generalization under challenging, non-stationary circumstances.

The initial findings of this study demonstrate that modular adaptivity can simultaneously increase a dynamic pipeline's recovery rate and robustness while also improving its overall accuracy; as a consequence, it has satisfied the fundamental requirements for a deployable spam filter under the ongoing evolution of new threats. Any one of the adaptive modules is harmful to the system as a whole, and including all of them minimizes performance changes when the content structure and high-noise labels change, according to controlled ablation tests. The aforementioned findings have some implications for the next generation of email security; only models that can handle drift explicitly and execute online updates will be able to keep up with new spam strategies, advances in hostile content, and organic shifts in user communication patterns.

There are still certain shortcomings. The study will employ a combination of controlled drift injections and historical datasets. novel threat vectors, including coordinated large-scale AI-driven attacks or novel generative spam, will necessitate additional validation and architectural enhancements. Real-time updates are still not appropriate for high-frequency or resource-constrained settings, despite being computationally efficient in this implementation. Expanding to multi-modal signals, facilitating cross-lingual adaptation, and including adaptive self-tuning in live, federated deployment scenarios will be the main goals of future optimizations. The aforementioned findings have strongly supported the creation of a large-scale, adaptable, and reliable email threat detection system in the future under evolving circumstances.

Author Contributions

Jarosław Jędras contributes to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, project administration, and funding acquisition. Przemysław Seweryn Grzelak contributes to methodology, software, validation. All authors have read and agreed with the manuscript before its submission and publication.

Funding

This research received no specific financial support from any funding agency.

Institutional Review Board Statement

Not applicable.

References

[1] Henke, M., dos Santos, E. M., Souto, E., & Santin, A. O. (2021). Spam Detection Based on Feature Evolution to Deal with Concept Drift. J. Univers. Comput. Sci., 27(4), 364-386. https://doi.org/10.3897/jucs.66284

[2] Filali, A., Shorfuzzaman, M., Abdellaoui Alaoui, E. A., Merras, M., Es-Sabry, M., Berrajaa, A., ... & Yousef, A. (2026). Cross-lingual SMS spam detection using GAN-based augmentation for imbalanced datasets. Scientific Reports. https://doi.org/10.1038/s41598-026-37769-4

[3] Ganguly, B., & Aggarwal, V. (2023). Online federated learning via non-stationary detection and adaptation amidst concept drift. IEEE/ACM Transactions on Networking, 32(1), 643-653. https://doi.org/10.1109/TNET.2023.3294366

[4] Roshan, M. K., & Zafar, A. (2024). Boosting robustness of network intrusion detection systems: A novel two-phase defense strategy against untargeted white-box optimization adversarial attack. Expert Systems with Applications, 249, 123567. https://doi.org/10.1016/j.eswa.2024.123567

[5] Li, Z., Huang, C., & Jia, X. (2025). MFFURL: Multi-modal Feature Fusion-Based Approach for Malicious URL Detection. Computer Networks, 111898. https://doi.org/10.1016/j.comnet.2025.111898

[6] Yanni, G. S., Salah, H., & Maghraby, F. A. (2024, December). Outsmarting Spam: Resilient Model for Concept Drift and Evolving Threats. In 2024 34th International Conference on Computer Theory and Applications (ICCTA) (pp. 138-144). IEEE. https://doi.org/10.1109/ICCTA64612.2024.10974776

[7] Kumar Birthriya, S., Ahlawat, P., & Kumar Jain, A. (2024). An efficient spam and phishing Email filtering approach using deep learning and bio-inspired particle swarm optimization. International Journal of Computing and Digital Systems, 15(1), 1-11. https://doi.org/10.12785/ijcds/150144

[8] Suhaimee, M. N., Shakil, F., Khan, M. R. A. A., Faruq, M. O., Sultana, S. R., & Firdaus, S. (2025, October). Real-Time Incremental Transformer with Continual Learning for Adaptive Phishing Email Detection. In 2025 IEEE 2nd International Conference on Computing, Applications and Systems (COMPAS) (pp. 1-6). IEEE. https://doi.org/10.1109/COMPAS67506.2025.11381774

[9] Naseeb, S., Ramzan, S., Raza, A., Hashmi, M. S. A., Gu, Y., Syafrudin, M., & Fitriyani, N. L. (2025). Website Phishing Attack Detection Using Innovative Meta Learning Based Ensemble Approach. IEEE Access. https://doi.org/10.1109/ACCESS.2025.3610961

[10] Hashmi, E., Yayilgan, S. Y., Hameed, I. A., Yamin, M. M., Ullah, M., & Abomhara, M. (2024). Enhancing multilingual hate speech detection: From language-specific insights to cross-linguistic integration. IEEE Access, 12, 121507-121537. https://doi.org/10.1109/ACCESS.2024.3452987

[11] Shyaa, M. A., Ibrahim, N. F., Zainol, Z. B., Abdullah, R., & Anbar, M. (2025). Reinforcement learning-based voting for feature drift-aware intrusion detection: An incremental learning framework. IEEE Access. https://doi.org/10.1109/ACCESS.2025.3544221

[12] Liu, J., Guo, F., Gao, H., Huang, Z., Zhang, Y., & Zhou, H. (2021). Image classification method on class imbalance datasets using multi-scale CNN and two-stage transfer learning. Neural Computing and Applications, 33(21), 14179-14197. https://doi.org/10.1007/s00521-021-06066-8

[13] Salman, M., Ikram, M., & Kaafar, M. A. (2024). Investigating evasive techniques in sms spam filtering: A comparative analysis of machine learning models. IEEE Access, 12, 24306-24324. https://doi.org/10.1109/ACCESS.2024.3364671

[14] Hovakimyan, G., & Bravo, J. M. (2024). Evolving strategies in machine learning: a systematic review of concept drift detection. Information, 15(12), 786. https://doi.org/10.3390/info15120786

[15] Zhao, Q., Gao, T., & Guo, N. (2023). Document-level relation extraction based on sememe knowledge-enhanced abstract meaning representation and reasoning. Complex & Intelligent Systems, 9(6), 6553-6566. https://doi.org/10.1007/s40747-023-01084-6

[16] Jáñez-Martino, F., Alaiz-Rodríguez, R., González-Castro, V., Fidalgo, E., & Alegre, E. (2023). A review of spam email detection: analysis of spammer strategies and the dataset shift problem. Artificial Intelligence Review, 56(2), 1145-1173. https://doi.org/10.1007/s10462-022-10195-4

[17] Alam, S., Jameel, A., Parveen, Z., & Alnfrawy, E. (2025). SHRED: An Ensemble-Based Machine Learning Model to Sift Email Messages for Real-Time Spam Detection. IEEE Access. https://doi.org/10.1109/ACCESS.2025.3605850

[18] Tusher, E. H., Ismail, M. A., Rahman, M. A., Alenezi, A. H., & Uddin, M. (2024). Email spam: A comprehensive review of optimize detection methods, challenges, and open research problems. IEEE Access, 12, 143627-143657. https://doi.org/10.1109/ACCESS.2024.3467996

[19] Chen, Z., Wang, Z., Zhou, Y., Liu, F., Liu, Y., Leng, T., & Zhu, H. (2024). A method for recovering adversarial samples with both adversarial attack forensics and recognition accuracy. Computers & Security, 144, 103987. https://doi.org/10.1016/j.cose.2024.103987

[20] Rashid, N., Mehmood, R., Alqurashi, F., Alqahtany, S., & Corchado, J. M. (2024). Asad: A meta learning-based auto-selective approach and tool for anomaly detection. IEEE Access, 13, 4341-4367. https://doi.org/10.1109/ACCESS.2024.3524908

[21] Saleem, S., Islam, Z. U., Hasan, S. S. U., Akbar, H., Khan, M. F., & Ibrar, S. A. (2025). Spam email detection using long short-term memory and gated recurrent unit. Applied Sciences, 15(13), 7407. https://doi.org/10.3390/app15137407

[22] Al-Kabbi, H. A., Feizi-Derakhshi, M. R., & Pashazadeh, S. (2023). Multi-type feature extraction and early fusion framework for sms spam detection. IEEE Access, 11, 123756-123765. https://doi.org/10.1109/ACCESS.2023.3327897

[23] Xia, Z., Liang, S., Wu, D., & Lv, S. (2026). Advancing Cross-Language Information Retrieval Through Shared Semantic Models: Applications in Public Cultural Resources. Applied Sciences, 16(9), 4158. https://doi.org/10.3390/app16094158

[24] Pelosi, D., Cacciagrano, D., & Piangerelli, M. (2025). Explainability and interpretability in concept and data drift: a systematic literature review. Algorithms, 18(7), 443. https://doi.org/10.3390/a18070443

[25] Jaffal, N. O., Alkhanafseh, M., & Mohaisen, D. (2025). Large language models in cybersecurity: A survey of applications, vulnerabilities, and defense techniques. AI, 6(9), 216. https://doi.org/10.3390/ai6090216

[26] Rassam, M. A., & Shaddad, R. (2026). Analysis of Robustness and Interpretability of Multinomial Naïve Bayes and Tiny Text CNN Models for SMS Spam Detection Under Adversarial Attacks. Information, 17(5), 408. https://doi.org/10.3390/info17050408

[27] Chen, Q., Wang, Z., Chen, J., Yan, H., & Lin, X. (2023). Dap-FL: Federated learning flourishes by adaptive tuning and secure aggregation. IEEE Transactions on Parallel and Distributed Systems, 34(6), 1923-1941. https://doi.org/10.1109/TPDS.2023.3267897

[28] Zhai, G., Yang, B., Sun, R., Liang, Z., Fang, K., & Li, Z. (2025, December). Multimodal Spam Detection Based on Multi-View Complementary Hybrid Feature Fusion. In 2025 IEEE International Conference on Blockchain Technology and Information Security (ICBCTIS) (pp. 1-8). IEEE. https://doi.org/10.1109/ICBCTIS66509.2025.11387473

[29] McCarthy, A., Ghadafi, E., Andriotis, P., & Legg, P. (2022). Functionality-preserving adversarial machine learning for robust classification in cybersecurity and intrusion detection domains: A survey. Journal of Cybersecurity and Privacy, 2(1), 154-190. https://doi.org/10.3390/jcp2010010

[30] Wang, Y., Xu, Y., Liu, Z., Liu, S., & Wu, Y. (2025). Research on Lightweight Dynamic Security Protocol for Intelligent In-Vehicle CAN Bus. Sensors, 25(11), 3380. https://doi.org/10.3390/s25113380