A Comparative Study of Metadata and Text-Based Features Using Stacked SVMs for Telegram Spam Detection

Nor Azman Mat Ariff[1*], Mohd Zaki Mas'ud[2], Mohd Najwan Md Khambari[1], Muhammad Shahkhir Mozamir[1], Mohd Fairuz Iskandar Othman[1]

[1] Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka, Malaysia

[2] Faculty of Artificial Intelligence and Cyber Security, Universiti Teknikal Malaysia Melaka, Malaysia

DOI: https://doi.org/10.47772/IJRISS.2026.101000785

Received: 27 October 2026; Accepted: 01 November 2026; Published: 18 November 2026

ABSTRACT

Telegram is frequently targeted by spammers because it supports large groups, allowing scammers to send a single message to a vast number of recipients. Spam tactics such as obfuscated text, intentional misspellings, and the substitution of symbols for letters or numbers can bypass filters that rely on complete, standard words. While Support Vector Machine (SVM) classifiers are well suited to small or high dimensional datasets such as spam text messages, kernel choice is critical because performance depends on the underlying data distribution. This study compared four feature representations, namely metadata, Bag of Words (BoW), word n-grams, and character n-grams, using SVMs with linear and RBF kernels. It also examined a combined model that integrated the four representations through stacking. During preprocessing, duplicate and invalid messages were removed, and the cleaned data were partitioned into training and testing sets using a stratified split. Undersampling was applied only to the training set, while the test set retained its original class imbalance to reflect real world conditions. Each SVM model was tuned using five fold grid search, evaluated with ten fold cross validation, and finally tested on the held out test set. The stacking approach used the decision scores of the four base models, with logistic regression as the meta learner. The character n-gram model with an RBF kernel was the best individual classifier, achieving an accuracy of 95.69% and a spam class F1 score of 92.85%. Stacking all four models achieved an accuracy of 96.28% and a spam F1 score of 93.84%, improvements of 0.59 and 0.99 percentage points, respectively. Removing metadata eliminated these gains in accuracy, while combining character n-grams with metadata yielded an accuracy of 96.06%. Character level features were the most effective individual representation, whereas metadata, which is readily available, provided a complementary perspective that modestly improved spam detection on Telegram.

Keywords: Character n-grams, metadata features, stacking ensemble, Support vector machine, Telegram spam detection

1. INTRODUCTION

Nowadays, Telegram has become a less trusted platform due to its susceptibility to spam attacks. Common spam attacks include phishing, the distribution of malware, and the takeover of Telegram accounts, incidents that can lead to users losing money, having their personal data compromised, or even losing control over their accounts. While existing spam classifiers are generally effective in detecting basic spam, whether via email, SMS, or Telegram, they often misclassify more sophisticated spam messages. Relying on only a single type of feature, as seen in studies Hassan and Wajiro (2025), and Hassan et al. (2024), limits a classifier's effectiveness, as no single feature type is able to address all classification challenges. Therefore, an adaptive solution that leverages multiple types of features, such as a combination of metadata based and text based features, needs to be developed to detect and mitigate spam in real time.

Despite advancements in machine learning, there remains a significant gap in the literature regarding the comparative effectiveness of metadata versus text based features for spam detection on Telegram. Existing models tend to focus on either lexical indicators or sender information, rarely integrating both into a unified

framework. Furthermore, the use of ensemble learning, specifically stacked Support Vector Machines, has yet to be thoroughly explored in this domain, leaving the question of its potential to enhance the accuracy and robustness of detection unanswered.(Xu et al., 2026; Xu et al., 2025; Skaržauskienė et al., 2026).

The integration of metadata features, such as character length, word count, average word length, and maximum word length, with text-based attributes, including lexical and semantic indicators, is increasingly recognised as a crucial element of effective spam detection. Combining these features not only enhances detection accuracy

but also strengthens resilience against sophisticated attack tactics, which are becoming more prevalent on platforms such as Telegram. This holistic approach is essential for developing scalable spam detection systems capable of operating in environments with high data-processing rates. (Dargahi Nobari et al., 2021; Verma & Sadana, 2025).

Stacked SVM classifiers offer a promising solution by combining the interpretability and computational efficiency of traditional SVMs with the strengths of ensemble learning. Their margin based learning capability and robustness against adversarial attacks make them particularly well suited for the operational context of Telegram, where deep learning models might be hindered by resource constraints and a lack of transparency.(Adnan et al., 2024; Chiu & Huang, 2007; Rączka & Plichta, 2026).

2. RELATED WORK

2.1 Metadata Features in Spam Detection

Metadata features have long served as a crucial component in spam detection across various messaging platforms, encompassing information such as sender details, message structure, and indicators of virality. On Telegram, these features are particularly valuable given the platform's use of a broadcasting system based on channels and real time message dynamics. Research indicates that metadata attributes, such as character length, word count, average word length, and maximum word length, provide contextual signals that complement lexical analysis, thereby enabling more accurate and adaptive spam detection (Dargahi Nobari et al., 2021; Verma & Sadana, 2025; Gourshettiwar et al., 2024; Şenol et al., 2026). Feature selection methods, such as LASSO and Mutual Information, further enhance classifier performance by isolating the most informative metadata features, while preprocessing techniques, including normalization and anonymization, improve generalization capabilities and reduce bias (Şenol et al., 2026).

2.2 Text-Based Feature Extraction and Classification

Text based feature extraction methods, including TFIDF, bag of words, and advanced embeddings such as Word2Vec, FastText, and contextual transformers, have been widely employed in spam detection. While traditional approaches such as TFIDF and bag of words remain effective for spam in SMS and email, the more complex and context dependent nature of spam content on Telegram necessitates more sophisticated techniques. Recent studies indicate that hybrid frameworks, which combine lexical, semantic, and contextual features, deliver superior performance and reduce classification errors, specifically false positives and false negatives(Xu et al., 2026; Kharat et al., 2026; Subhashini et al., 2024; Chaudhary & Jain, 2026). Deep learning models, such as LSTMs and transformer based architectures, are highly effective at capturing semantic nuances, but these models often require significant computational resources and large datasets for training.

2.3 Stacked SVM Classifiers in Spam Detection

Ensemble and stacking techniques have emerged as effective tools for improving the accuracy of spam detection. Specifically, stacked SVM classifiers leverage multiple layers of margin based learning to optimize precision and recall. However, the base classifiers that comprise the ensemble must be diverse, given that an ensemble is only effective when its members exhibit different error patterns. The predictions from these members are fed into a meta-classifier, which then learns how to combine them. Comparative analysis indicates that stacked SVMs outperform single SVMs and are comparable to deep learning models in resource constrained, real time environments; they offer a balance between computational efficiency and detection robustness(Adnan et al., 2024; Rączka & Plichta, 2026; Ying, 2012; Poojashri et al., 2025). Systematic kernel selection, e.g., RBF, linear,

and hyperparameter tuning, using metaheuristic algorithms such as WOA or SPO, are crucial for maximizing performance, while adversarial training enhances resilience against continually evolving spam tactics(Koch et al., 2012).

3. METHODOLOGY

The methodology followed five stages as in Figure 1. The Telegram dataset was cleaned and partitioned. Four

feature representations were then extracted and classified with tuned linear and RBF SVMs. Their outputs were combined through a stacking ensemble, and all models were evaluated on a held-out test set using imbalance-aware metrics.

3.1 Dataset and Data Partitioning

The Telegram spam dataset was obtained from Kaggle and consisted of text messages labelled as either legitimate (ham) or spam. Before modelling, the data were screened to remove duplicate and invalid messages, so that repeated entries could not appear in both the training and test sets and inflate performance. The cleaned dataset was then divided into training and test subsets using a stratified 80/20 split, which preserved the original class proportions in both. Because spam was the minority class, random undersampling was applied to the training subset only, which balanced the two classes and reduced bias towards the majority class during learning. The test subset was never resampled, so it kept its natural, imbalanced class distribution and gave a realistic estimate of performance in deployment. Restricting resampling to the training data also prevented information from the test set from leaking into model development.

3.2 Feature Representations

Four feature representations were constructed, each capturing a different aspect of a message.

- **Metadata**: a compact, low-dimensional set of structural statistics, namely message length, the presence of digits and URLs, uppercase letters, punctuation, non-ASCII characters and currency symbols. These describe the form of a message rather than its content.
- **Bag of Words (BoW)**: each message was represented by the frequency of the individual words it contained, ignoring word order and grammar. This gave a high-dimensional but sparse representation of lexical content.
- **Word n-grams**: each message was represented by single words (unigrams) and pairs of consecutive words (bigrams). This captured some local word order and short phrases, but it enlarged the feature space considerably compared with BoW.
- **Character n-grams**: sub-word sequences intended to capture misspellings, obfuscated words, symbols and mixed-language text, which word-based features treat as separate tokens.

3.3 Base Classifiers and Hyperparameter Tuning

Every feature representation was classified with linear and RBF-kernel support vector machines (SVMs) under one identical procedure, so that differences in performance reflected the features and kernels rather than the experimental design. This procedure had three steps.

1. **Tuning:** hyperparameters were selected by five-fold stratified grid search on the balanced training set, with macro-F1 as the selection criterion.

2. **Validation:** the tuned model was assessed by ten-fold stratified cross-validation.

3. **Testing:** the model was evaluated once on the held-out test set.

3.4 Stacking Ensemble

Four base models were combined: metadata, BoW, word n-grams and character n-grams. Hyperparameters were taken from the single-classifier experiments. Out-of-fold decision scores were generated by five-fold cross-validation on the balanced training set, so the meta-learner never saw scores from a model that had been trained

on the same sample. A logistic regression meta-learner was trained on these scores and learned the weight each base model deserved. The Pearson correlations between out-of-fold scores were examined to measure redundancy between models. High correlations indicated overlapping information, and low correlations indicated complementary views.

Three ensembles were then compared with the best single model:

- all four base models;
- the three text-based models without metadata;
- the best text-based model combined with metadata.

3.5 Evaluation Metrics and Statistical Testing

Because the test set was imbalanced, accuracy alone was not sufficient. Spam-class precision, recall and F1-score and were therefore reported alongside accuracy. Precision reflected false alarms on legitimate messages, and recall reflected missed spam.

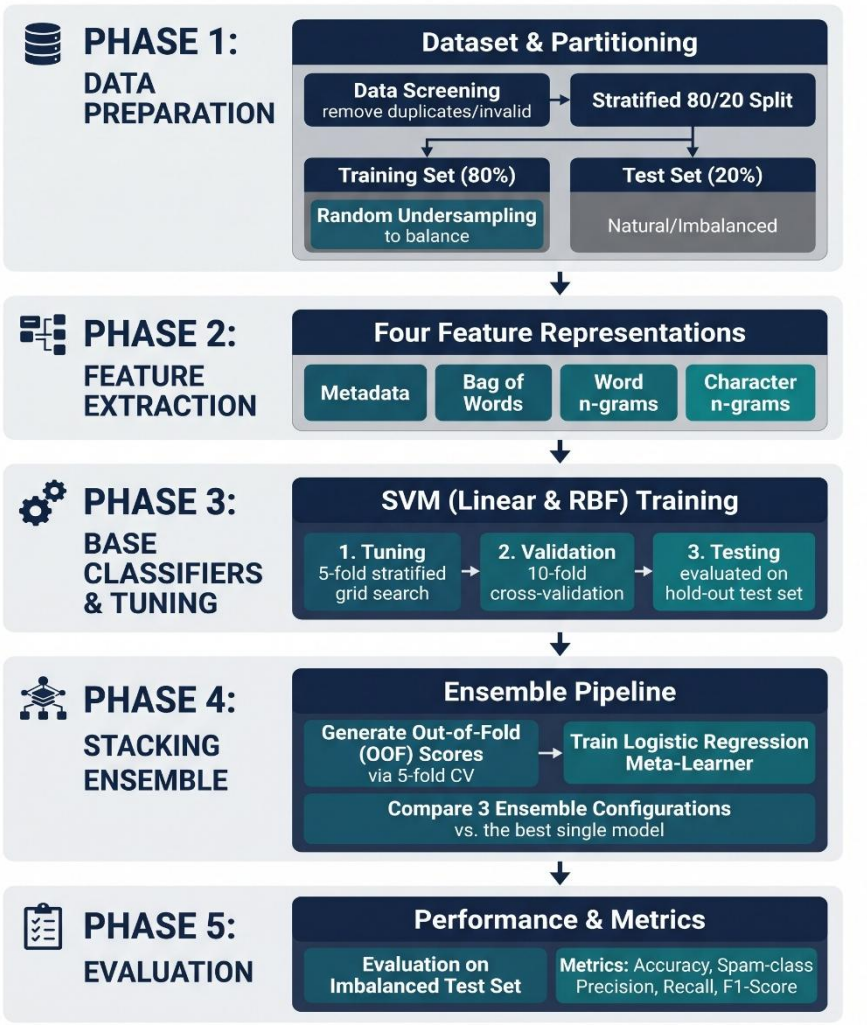


Figure 1 : Proposed Stacking Ensemble Methodology for Telegram Spam Detection

4. RESULTS

4.1 Experimental Setup

The Telegram spam dataset was obtained from Kaggle and contains 20,348 messages. After 68 duplicate or invalid messages were removed, 20,280 messages remained (14,271 ham and 6,009 spam). A stratified 80/20

split produced a training set of 16,224 messages and a test set of 4,056 messages. Random undersampling was applied to the training set only, yielding 9,610 balanced messages (4,805 per class). The test set was never resampled. Every feature representation was evaluated with linear and RBF-kernel support vector machines (SVMs) under the same procedure: five-fold stratified grid search (macro-F1), then ten-fold stratified cross-validation (CV), then a single evaluation on the held-out test set.

4.2 Single-Model Results

As shown in Table 1, the character n-gram using an RBF kernel is the best model across three metrics, namely accuracy, spam precision and spam F1, with 95.69%, 91.18% and 92.85%, respectively. It is only 0.09% below the character n-gram using a Linear kernel for the spam recall metric, recording 94.59% compared to the linear kernel's 94.68%. Character n-grams may be effective at classifying spam messages that contain obfuscated words, misspellings, symbols, and mixed-language text as they still capture these patterns while word-based features treat each variant as a different word.

Metadata is the weakest model compared to the others, recording an accuracy of 82.30–85.87%, a spam precision of 71.19–74.44%, a spam recall of 67.64–79.70%, and a spam F1-score of 69.37–76.98%. It identifies the structural characteristics of a message, such as its length, the presence of digits, and URLs, rather than its actual content. Text-based features offer a higher level of accuracy, with a margin of difference of approximately 8–10%. Nevertheless, metadata remains worth discussing due to its low computational cost, and its contribution within stacking methods can be examined separately.

The RBF kernel outperforms the linear kernel for most feature sets, with a largest improvement in metadata that is 3.57% for accuracy and 7.61% for spam F1. A modest performance improvement can be seen for the BoW feature set and n-gram characters. The BoW feature set recorded an increase in accuracy by 0.61% and spam F1 by 1.16% while the character n-gram feature set also increased by 0.60% and 0.89% for accuracy and spam F1 respectively. Nevertheless, almost the same performance for the n-gram feature set with the RBF kernel and the Linear kernel for accuracy and an increase of 0.02% for spam F1. Metadata is a low-dimensional compact feature set, so the use of non-linear boundaries is beneficial. High-dimensional text data that is sparse can almost be separated linearly; this is why the linear model gives almost equivalent performance.

Word n-grams exhibit high recall, at approximately 94%, but the lowest precision among text-based models, at around 86.8%. This indicate that they successfully detect most spam, but also misclassify a larger number of legitimate messages as spam. Character n-grams show similar performance, with a precision of approximately 89.4% to 91.2% and a recall of 94.6%, giving the most balanced performance. BoW displays the opposite pattern, with slightly higher precision, at around 90.1%, and lower recall, ranging from 89.9% to 92.4%.

The text-based models only differ by about 1.6% in terms of accuracy where the lowest accuracy is 94.06% and the highest accuracy is 95.69%. The McNemar's exact test showed that the character n-gram RBF performed significantly better than all other single classifiers ($p < 0.01$), with the narrowest margins against BoW RBF and character n-gram Linear.

Individual base classifiers exhibit complementary strengths. For instance, metadata features capture the structural characteristics of a message, such as its length, the use of digits, and the presence of URLs whereas text-based feature sets (e.g., BoW, word n-grams, and character n-grams) capture its lexical content. Their error profiles also differ: word n-grams achieve a high recall of approximately 94% but the lowest precision among the text-based models (around 86.8%), while BoW demonstrates higher precision (90.1%) but lower recall (89.9–92.4%). Given that these classifiers rely on different information and make partly different errors, combining their outputs may correct errors made by any single model. This forms the basis for the stacking stage, in which a logistic regression meta-learner takes the SVM decision scores of the single classifiers as input and learns how much weight each one deserves.

Table 1. Performance (%) of single base classifiers

Features and **Kernel**	Accuracy	Spam Precision	Spam Recall	Spam F1

Metadata Linear	82.3	71.19	67.64	69.37
Metadata RBF	85.87	74.44	79.7	76.98
BoW Linear	94.11	90.16	89.93	90.05
BoW RBF	94.72	90.1	92.35	91.21
word n-gram Linear	94.06	86.93	94.09	90.37
word n-gram RBF	94.06	86.82	94.26	90.39
char n-gram Linear	95.09	89.4	**94.68**	91.96
char n-gram RBF	**95.69**	**91.18**	94.59	**92.85**

4.2 Stacking Ensemble Results

Four base models, derived from four distinct feature sets, were combined: metadata (RBF), BoW (RBF), word n-grams (linear), and character n-grams (RBF). All models utilize the RBF kernel, except word n-grams model. Both kernels for word n-grams models yield nearly identical performance (94.06%) but linear kernel was chosen because it is simpler and approximately 3.4 times faster to tune. Hyperparameters were determined through experiments on single classifiers. Out-of-fold decision scores (using five-fold cross-validation) were generated based on a balanced training set. A logistic regression meta-learner was trained using these decision scores. Table 2 shows the correlation of out-of-fold decision scores between models. This is used to identify how similar the models' information is. Very high values signal redundancy. The correlations among the scores from the three text-based views range from 0.894 to 0.957, whereas the correlations between the metadata and the text-based models fall within the 0.650 to 0.680 range, suggesting that metadata provides a different view of the messages.

Table 2. Correlation of out-of-fold scores

	metadata	BoW	word n-grams	character n-grams
metadata	1	0.68	0.65	0.671
BoW	0.68	1	0.903	0.894
word n-gram	0.65	0.903	1	0.957
char n-gram	0.671	0.894	0.957	1

Table 3 presents the ensemble results on the test set. Stacking the four models improved accuracy from 95.69% to 96.28%, and the spam F1-score from 92.85% to 93.84%, compared with the best single model (character n-grams). Removing the metadata model eliminated the accuracy gain, returning it to 95.69%, with a negligible change in spam F1-score (92.91%). Conversely, combining only character n-grams and metadata yielded better classification results, 96.06% which outperformed both the best single model and the stacking without metadata.

Table 3 : Performance (%) of stacking ensembles compared with the best single model

Model	Accuracy	Spam precision	Spam recall	Spam F1
Char n-grams (best single)	95.69	91.18	94.59	92.85
Stack: all four	**96.28**	**92.01**	**95.76**	**93.84**
Stack: without metadata	95.69	90.59	95.34	92.91
Stack: char + metadata	96.06	91.48	95.59	93.49

5. DISCUSSION

Character n-grams represent the most robust single model. A plausible explanation is that sub-word patterns can capture spelling errors, obfuscation, and partial word matches that are missed by word-based features such as BoW and word n-grams. In fact, a linear character n-grams model, achieving 95.09% accuracy, outperforms the best word-level model, which is BoW with an RBF kernel at 94.72%, while requiring less tuning time than an RBF-based model, making it an efficient and practical choice.

Incorporating word bigrams into the *word n-gram* model increases the dimensionality to 72,337 features, compared with the 17,512 features used by the *BoW* model, and this adds to the tuning time without yielding any improvement in classification performance.

As expected, metadata features alone are weaker than text-based features, yet they still contain useful information. Simple formatting statistics, such as the use of uppercase letters, punctuation, non-ASCII characters, currency symbols, and message length, are capable of distinguishing between the classes to a meaningful degree. The clear advantage of an RBF kernel over a linear kernel for metadata, with a spam F1-score of 76.98% versus 69.37%, suggests that these statistics interact in a non-linear fashion, whereas high-dimensional, sparse text data tends to be more linearly separable.

Three stacking ensembles were developed and compared with the best single model, the character *n-gram* classifier. This classifier achieved an accuracy of 95.69% and a spam F1-score of 92.85%. Stacking all four base models yielded the best overall results, with an accuracy of 96.28% and a spam F1-score of 93.84%. This represents an improvement of 0.59% in accuracy and 0.99% in the spam F1-score compared with the best single model. Every metric showed improvement, spam precision increased by 0.83%, and spam recall increased by 1.17%. This indicates that the ensemble successfully reduced both missed spam cases and legitimate messages that were incorrectly flagged as spam.

The role of metadata becomes apparent when the stacking ensembles are compared with one another. Removing the metadata model from the ensemble did not alter the accuracy, which remained at 95.69%, and increased the spam F1-score by only 0.06%. It also decreased spam precision by 0.59%, from 91.18% to 90.59%, even though recall increased by 0.75%. Consequently, this stacking ensemble detected more spam but at the cost of an increased number of false alarms, while the overall balance remained virtually unchanged. Conversely, a combination consisting solely of the character *n-gram* model and the metadata model yielded an accuracy of 96.06% and a spam F1-score of 93.49%, which are improvements of 0.37% and 0.64%, respectively. This compact stacking ensemble, consisting of two models, outperforms the three-model text-based ensemble on every metric, although the two achieved nearly identical spam recall, at 95.59% and 95.34%. This pattern indicates that metadata provides information not captured by text-based models, even though metadata is less effective on its own as a standalone classifier.

6. CONCLUSION

This study compares individual SVM classifiers built using metadata, BoW, word n-gram, and character n-gram as features, and subsequently combines them via a stacking ensemble technique. The character n-gram model with an RBF kernel emerged as the best individual classifier, achieving an accuracy of 95.69%, a spam precision of 91.18%, a spam recall of 94.59%, and a spam F1-score of 92.85%. Its strength likely stems from its ability to capture spelling errors, obfuscation techniques, and partial word matches, which are aspects that word based features fail to capture. When used in isolation, metadata serves as the weakest feature set, yet it contains useful structural information. In general, the RBF kernel outperforms the linear kernel, particularly for metadata, whereas both perform comparably when applied to sparse text features.

Stacking these four base models yields the best overall results, with an accuracy of 96.28% and a spam F1-score of 93.84%, which is an improvement of 0.59 and 0.99 percentage points, respectively, over the best individual model. Metadata plays a crucial role in this improvement, as its removal causes the accuracy to revert to 95.69%, while a combination of character n-grams and metadata alone achieves 96.06%. The low correlation between metadata and text based models suggests that metadata provides a complementary perspective on the messages.

Overall, the application of a stacking technique that incorporates metadata offers a modest improvement in spam detection on Telegram.

REFERENCES

1. Adnan, M., Imam, M. O., Javed, M. F., & Murtza, I. (2024). Improving spam email classification accuracy using ensemble techniques: a stacking approach. *International Journal of Information Security*, *23*(1), 505–517. https://doi.org/10.1007/s10207-023-00756-1
2. Chaudhary, A., & Jain, S. (2026). Enhancing text classification using hybrid embeddings and advanced machine learning techniques. *International Journal of System Assurance Engineering and Management*, *17*(1), 214–229. https://doi.org/10.1007/s13198-025-02950-x
3. Chiu, C.-Y., & Huang, Y.-T. (2007). Integration of support vector machine with naïve Bayesian classifier for spam classification. *Proceedings - Fourth International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2007*, *1*, 618–622. https://doi.org/10.1109/FSKD.2007.366
4. Dargahi Nobari, A., Sarraf, M. H. K. M., Neshati, M., & Erfanian Daneshvar, F. (2021). Characteristics of viral messages on Telegram; The world's largest hybrid public and private messenger. *Expert Systems with Applications*, *168*. https://doi.org/10.1016/j.eswa.2020.114303
5. Gourshettiwar, P., Thakare, R. D., Duddalwar, P., Shete, R., Gundewar, S., & Verma, P. (2024). ShieldSMS Sentinel: Guardian of Your Digital Communication Bastion. *2024 2nd DMIHER International Conference on Artificial Intelligence in Healthcare, Education and Industry, IDICAIEI 2024*. https://doi.org/10.1109/IDICAIEI61867.2024.10842929
6. Hassan, A., & Wajiro, M. A. (2025). BES-SVM: Enhanced machine learning algorithm for Telegram spam classification. *FUOYE Journal of Engineering and Technology, 10*(4), 586–589. https://doi.org/10.4314/fuoyejet.v10i4.2
7. Hassan, A., Ayuba, Y., Wajiro, M. A., & Ahmad, M. Z. (2024). Machine learning algorithms for Telegram spam filtering. *FUDMA Journal of Sciences, 8*(6), 170–176. https://doi.org/10.33003/fjs-2024-0806-2799
8. Kharat, R., Bhattarai, M., Chavan, A., Chaudhary, R. N., & Bhore, A. (2026). Spam Detection in Text Messages and Emails: A Natural Language Processing Approach. *International Conference on Emerging Trends and Innovations in ICT, ICEI 2026*. https://doi.org/10.1109/ICEI65890.2026.11448007
9. Koch, P., Bischl, B., Flasch, O., Bartz-Beielstein, T., Weihs, C., & Konen, W. (2012). Tuning and evolution of support vector kernels. *Evolutionary Intelligence*, *5*(3), 153–170. https://doi.org/10.1007/s12065-012-0073-8
10. Poojashri, P., Shopika, S., & Pavithra, S. (2025). Enhancing Online Transaction Security Through Comprehensive Fraud Detection Using Machine Learning. *2025 International Conference on Data Science, Agents and Artificial Intelligence, ICDSAAI 2025*. https://doi.org/10.1109/ICDSAAI65575.2025.11011564
11. Rączka, G., & Plichta, A. (2026). Spam and Phishing Detection with Adversarial Attack Robustness Analysis Using Machine Learning Models. In S. F., D. F., S. F., P. M., A.-B. K., & I. M. (Eds.), *Proceedings - European Council for Modelling and Simulation, ECMS* (Vols. 2026-June, pp. 311–320). European Council for Modelling and Simulation. https://www.scopus.com/inward/record.uri?eid=2-s2.0-105042236606&partnerID=40&md5=0edaadea60f6f0ff0393ba4c17bff2f1
12. Rumapea, H., Manalu, D. R., & Rumapea, Y. Y. P. (2025). Interpretable Deep Learning for Enhanced AI Trust and Clarity. *Journal of Artificial Intelligence and Technology*, *5*, 345–353. https://doi.org/10.37965/jait.2025.0748
13. Şenol, T., Takcı, H., & Şeker, A. (2026). SMS spam detection via a systematic machine learning pipeline: Feature engineering, hyperparameter optimizing, scaling, and ensemble learning. *Journal of Engineering Research (Kuwait)*, *14*(3), 2941–2952. https://doi.org/10.1016/j.jer.2026.05.020
14. Skaržauskienė, A., Mačiulienė, M., Gulevičiūtė, G., Zelenkauskaitė, A., Diržytė, A., & D'Antonio Maceiras, S. (2026). MALICIOUS CREATIVITY IN TELEGRAM'S (SOFTWARE) ANTI-VACCINATION ECOSYSTEM: PROFILING ACTORS AND EARLY MISINFORMATION DYNAMICS. *Creativity Studies*, *19*(1), 132–141. https://doi.org/10.3846/cs.2026.22247
15. Subhashini, G., Mahalakshmi, G., Ashik, H. M., & Duresh, B. N. (2024). Advanced SMS Spam

Detection Using Integrated Feature Extraction. *Proceedings of the 4th International Conference on Ubiquitous Computing and Intelligent Information Systems, ICUIS 2024*, 780–785. https://doi.org/10.1109/ICUIS64676.2024.10867034

16. Verma, A. R. K., & Sadana, S. (2025). Textual, Non-Textual, and Hybrid Feature Engineering for SMS Spam Classification. *IEEE Access*, *13*, 176901–176914. https://doi.org/10.1109/ACCESS.2025.3620751
17. Xu, S., Ding, Z., Wei, Z., Yang, C. P., Li, Y., Chen, X., & Wang, H. (2026). Beyond keywords: modeling the semantic complexity of deceptive communication on instant messaging platforms. *Journal of Computational Social Science*, *9*(2). https://doi.org/10.1007/s42001-026-00461-8
18. Xu, S., Ding, Z., Wei, Z., Yang, C. P., Li, Y., Chen, X., & Wang, H. (2025). A Comparative Analysis of Deep Learning and Machine Learning Approaches for Spam Identification on Telegram. *2025 6th International Conference on Computer Communication and Network Security, CCNS 2025*, 50–55. https://doi.org/10.1109/CCNS66849.2025.11338070
19. Yadav, A., Yadav, R., & Kiruthika, M. (2025). SMS Spam Detection with NLP and Ensemble Learning for Enhanced Classification. *2025 International Conference on Computing Technologies and Data Communication, ICCTDC 2025*. https://doi.org/10.1109/ICCTDC64446.2025.11159039
20. Ying, K. (2012). Study on image multiple classification based on boosting method. *Journal of Convergence Information Technology*, *7*(13), 58–65. https://doi.org/10.4156/jcit.vol7.issue13.7