BULLETIN of Cherkasy State Technological University

Journal homepage: https://bulletin-chstu.com.ua/en

Vol. 30 No. 3. 2025

UDC 004.056.5:004.42:004.942.5
DOI: 10.62660/bcstu/3.2025.121

Adaptive hybrid SMS spam detection system with user feedback-based self-learning

Ihor Liakh

Doctor of Technical Sciences, Professor
Uzhhorod National University
88000, 3 Narodna Sqr., Uzhhorod, Ukraine
https://orcid.org/0000-0001-5417-9403

Andriy Chorniy*

PhD in Technical Sciences, Associate Professor
Cherkasy State Technological University
18006, 460 Shevchenka Blvd., Cherkasy, Ukraine
https://orcid.org/0000-0002-0989-7112

Oksana Lutak

Master's Student
Uzhhorod National University
88000, 3 Narodna Sqr., Uzhhorod, Ukraine
https://orcid.org/0009-0002-0798-6928

Marian Tsenkner

Master's Student
Uzhhorod National University
88000, 3 Narodna Sqr., Uzhhorod, Ukraine
https://orcid.org/0009-0003-2752-7398

Abstract. This study presented a comprehensive approach to SMS spam detection based on a hybrid architecture that integrated local message processing algorithms with high-performance cloud-based deep learning models. This approach enabled a balance between classification accuracy and the privacy of processed messages. The objective of this study was to develop an intelligent hybrid SMS spam detection system capable of delivering high classification accuracy, maintaining up-to-date knowledge, enabling user personalisation, and adapting to new attack patterns. To achieve the study's objective, a comprehensive analytical approach was applied, combining a detailed review of scientific literature on SMS spam detection – including machine learning, neural networks, and hybrid methods – with empirical analysis. To implement classic machine learning models (Naïve Bayes, Logistic Regression, Random Forest), standard machine learning libraries were used, and for deep learning, frameworks that support recurrent neural networks, in particular Long Short-Term Memory and transformer architectures, were applied. The system was tested on the open SMS Spam Collection dataset using Accuracy (up to 0.98), F1-score (up to 0.95) and ROC-AUC (up to 0.98) metrics. Moreover, a system was developed to dynamically update knowledge based on user feedback, alongside a weighted framework designed to evaluate the trustworthiness of that feedback. During the study, a multi-level system was developed that performed initial classification on the user's device with the ability to delegate processing to a cloud module in cases

Article's History: Received: 28.03.2025; Revised: 01.08.2025; Accepted: 15.09.2025.

Suggested Citation:
Liakh, I., Chorniy, A., Lutak, O., & Tsenkner, M. (2025). Adaptive hybrid SMS spam detection system with user feedback-based self-learning. *Bulletin of Cherkasy State Technological University*, 30(3), 121-132. doi: 10.62660/bcstu/3.2025.121.

*Corresponding author

of uncertainty. Compared to basic approaches, the hybrid architecture demonstrated improved classification accuracy, reduced false positives and false negatives, and increased adaptability to changes in the structure of spam messages. Aggregation of suspicious messages in the cloud ensured effective retraining of models in cases of conceptual shift. The practical value of the results lies in the potential integration of the developed system into mobile platforms, as well as corporate information security tools, for the purpose of filtering SMS content and protecting end-users from social engineering

Keywords: natural language processing; long short term memory; architecture of spam; messages; metric

INTRODUCTION

The relevance of the study was determined by the rapid increase in the number of malicious Short Message Service (SMS) messages, which were used as tools of phishing, social engineering, and fraud. The complication of spam text structures, the emergence of disguised messages, and the use of bypassing techniques by attackers significantly reduced the effectiveness of traditional filtering methods. At the same time, the need for the development of intelligent systems grew, as they had to ensure high classification accuracy while preserving user data confidentiality. Increased requirements for the speed, adaptability, and scalability of message processing created the demand for new architectures that combined local processing with cloud computing and supported mechanisms of continuous knowledge updating in the dynamic environment of mobile threats.

Research devoted to the detection of mobile SMS spam, considerable attention was paid to the application of hybrid systems, text vectorisation methods, and the use of natural language processing (NLP). One of the promising directions was the use of federated learning, which allowed the preservation of user privacy without a loss in model accuracy. The scientists S. Vats *et al*. (2024) studied SMS spam classification within the framework of federated learning, emphasising the balance between model accuracy and the preservation of user privacy. In the proposed approach, message processing took place directly on the user's device, while the already trained models were combined without transferring personal data to a central server. The study results demonstrated the high efficiency of the method: classification accuracy reached 98.3% as early as the third training stage, which confirmed the stability and accuracy of the model while maintaining confidentiality. In addition, a gradual decrease in the loss function was observed from the first to the tenth epoch, which proved the effectiveness of the learning process. The authors gave insufficient attention to the issue of model scalability under limited computational resources and did not address the problems of concept drift and adaptation to dynamically changing spam patterns, which were critically important for the practical implementation of hybrid systems.

Hybrid SMS spam detection systems, which combined several machine learning (ML) algorithms, continued to demonstrate high adaptability to new attack scenarios. The researchers H. Baaqeel & R. Zagrouba (2020) investigated the problem of classifying SMS messages as spam or "ham" using ML methods. In their study, they proposed a hybrid system that integrated supervised and unsupervised learning algorithms to improve the accuracy of unwanted message detection. The authors concluded that the application of a combined approach enhanced the system's performance, particularly in terms of accuracy and the F-measure. Simultaneously, the study devoted insufficient attention to the aspects of system adaptability to new types of attacks and the possibility of dynamic knowledge updating, which were essential for counteracting the rapid evolution of spam techniques.

Researchers A. Al Maruf *et al*. (2023) developed an ensemble-based approach to the classification of SMS messages in Bengali (Bangla), which enables the detection of spam even with a relatively small training dataset. The study employed logistic regression, decision trees, support vector machines, and random forests as baseline models, with the results subsequently enhanced through ensemble techniques such as bagging, boosting, and stacking. The authors concluded that the ensemble methodology, particularly the XGBoost model, achieved the highest classification accuracy for Bengali SMS. However, the study provided insufficient discussion regarding the scalability of the proposed approach to larger datasets and other languages, as well as the robustness of the system against emerging types of spam, which limits the potential for its broader application.

The researchers M.A. Abid *et al*. (2022) focused on the importance of high-quality text feature preparation in classifying SMS messages as spam or "ham" using classical ML algorithms. In the study, the bag-of-words and term frequency-inverse document frequency (TF-IDF) methods were applied for feature construction, and over-sampling and under-sampling strategies were employed to balance the uneven dataset. The experimental results showed that the Random Forest algorithm delivered the highest classification accuracy, reaching 99% for both SMS and a spam email corpus. The authors concluded that proper feature engineering played a key role in the performance of the models. Concurrently, the study gave insufficient attention to the generalisation of the approach to multilingual corpora and to its effectiveness in handling dynamically changing data, which limited the practical applicability of the developed system.

The researchers D.A. Oyeyemi & A.K. Ojo (2024) proposed an approach to SMS spam detection based on

NLP methods combined with ML models. In the study, data preprocessing (removal of stop words, tokenisation) and vectorisation using Bidirectional Encoder Representations from Transformers (BERT) were implemented. Classification was carried out with several algorithms, including support vector machine (SVM), logistic regression, the Naïve Bayes classifier, gradient boosting, and random forest. The results showed that the combination of Naïve Bayes and BERT achieved the highest accuracy (97.31%) with minimal execution time (0.3 seconds), which indicated the high efficiency of the approach and a low rate of false positives. The authors concluded that the integration of BERT with classical algorithms was capable of significantly improving the quality of SMS spam filtering. Contemporaneously, the study devoted insufficient attention to the scalability of the solution for large datasets and to its applicability in multilingual environments, which might limit the practical use of the proposed system.

The researchers S. Gadde *et al*. (2021) investigated the application of ML and deep learning (DL) methods for SMS spam detection. In the study, the UCI dataset was used, on which the authors compared the performance of classical ML algorithms with recurrent neural networks using long short-term memory (LSTM). The experimental results showed that the LSTM model achieved an accuracy of 98.5%, outperforming the other tested algorithms, which confirmed the relevance of integrating DL into spam filtering systems. The authors concluded that the combination of ML and DL methods could significantly improve system effectiveness in real-world conditions. In parallel, the study gave limited consideration to the optimisation of the LSTM architecture for different corpus sizes and did not analyse the computational resource costs, which were important for the practical implementation of such models.

The additional difficulties in combating spam arose from the methods employed by attackers to bypass detection systems. M. Salman *et al*. (2024) conducted a comparative analysis of various ML models and contemporary anti-spam services using a new large dataset of over 68,000 SMS messages, of which 39% were spam. The study included both shallow ML methods and deep neural networks, as well as an analysis of the models' robustness to evolutionary and covert attacks by spammers. The results showed that most classical ML solutions and existing anti-spam services demonstrated insufficient effectiveness against bypass strategies. In tandem, the authors concluded that the combination of classical and modern models could enhance system resilience to such attacks, opening avenues for the development of more reliable SMS spam filters.

In turn, V.V. Kalyani *et al*. (2024) demonstrated the effectiveness of recurrent neural networks (RNN) for SMS spam classification. Using a dataset of 5,570 messages, the authors first addressed the class imbalance problem using the ADASYN method and applied TF-IDF vectorisation for text representation. Baseline ML algorithms, including Random Forest, achieved an accuracy of 92.3%, whereas RNN architectures significantly outperformed these results: LSTM reached 98%, Bi-LSTM 98.2%, and GRU 98.3%. The highest performance was delivered by a hybrid model combining Bi-LSTM and GRU, achieving 99% accuracy. The authors' contribution lay in demonstrating the advantages of DL models over classical ML approaches, as well as in the use of data balancing methods, which enabled maximum accuracy even with a limited dataset size. Therefore, the analysis of current literature confirmed the relevance of developing hybrid, adaptive, and privacy-oriented approaches to SMS spam detection, combining the advantages of ML, NLP, and continuous knowledge updates. The aim of this study was to develop an adaptive SMS spam detection system that integrates both local and cloud-based classification models, enabling continuous knowledge updates.

MATERIALS AND METHODS

The study was conducted within the framework of developing an intelligent system capable of effectively countering SMS spam with dynamic patterns, including phishing messages (smishing). The methodological foundation of the study was based on the principles of NLP, adaptive text data analysis, and dynamic knowledge updating. For empirical testing of the proposed approach's effectiveness, the publicly available SMS Spam Collection Dataset was used, containing 5,574 SMS messages that had been pre-classified as either spam or legitimate. The effectiveness of the proposed SMS spam detection system was evaluated through a series of experiments using publicly available datasets. The primary testing platform was the SMS Spam Collection Dataset, which contained 5,574 labelled examples of real text messages, including 4,827 legitimate (ham) messages and 747 spam messages (Almeida & Hidalgo, 2011).

During the text data preprocessing stage, standard operations were performed, including normalisation, removal of noise elements, tokenisation, and the transformation of messages into vector-based features using frequency-statistical methods (e.g., TF-IDF) and semantic models. At the local level, a pre-trained model was applied to instantly classify incoming messages as spam or legitimate. In cases of uncertainty or the detection of new patterns, the system could initiate forwarding the message to the cloud for re-analysis using a more resource-intensive but more accurate model. Contemporary DL methods were employed for this purpose, particularly LSTM models, which had demonstrated high effectiveness in text classification tasks (Gomaa, 2020). The training of such models was often based on the minimisation of a loss function, for example, the cross-entropy function:

$$CrossEntropy(p, q) = -\sum_n p_n \times ln(q_n), \quad (1)$$

where p – the true label and q – the predicted value from the model's output.

For the construction of classification models, both classical statistical algorithms (Naïve Bayes, Logistic Regression) and more complex ensemble and DL models (Random Forest, LSTM) were employed, allowing a comparative analysis of their effectiveness. Each model was evaluated using the primary metrics for binary classification: accuracy, precision, recall, F1-score, and ROC-AUC. These metrics enabled a comprehensive assessment of the models' ability to effectively detect SMS spam while minimising false-positive classifications. Subsequently, the vector representation was sent to a local filter implemented using a ML model (e.g., Naïve Bayes or Logistic Regression) according to the study of D. Bäckman (2019). For the Naïve Bayes method, the probability that a message belonged to a certain class C given the presence of features X is determined by the following formula:

$$P(C|X) = \frac{P(X|C)*P(C)}{P(X)}, \quad (2)$$

where $P(C|X)$ – the *posterior* probability of the class, $P(X|C)$ – the likelihood, $P(C)$ – the prior probability of the class, and P(X) is the probability of the features.

For Logistic Regression, the logistic function (sigmoid) was used, which transforms a linear combination of features z into a value within the range (0.1):

$$\Lambda(z) = \frac{1}{1+e^{-z}}, \quad (3)$$

where $z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n$, β_i – model coefficients, x_i – feature values.

The architecture of the proposed system provided for local preliminary classification of messages, with the option to forward uncertain cases to the cloud model for enhanced analysis. The cloud model employed DL methods and performed periodic retraining based on new data and user feedback. The study implemented a mechanism for dynamic knowledge updating, considering user feedback, its reliability, and changes in the statistical characteristics of the data over time. To ensure stability, change buffers, update parameter controls, and anomaly aggregation were used. Table 1 provides the structure of a representative sample used during model retraining. The sample includes different message types, their quantities, the reliability of user feedback associated with each type, and the assigned sample weights used to prioritise model updates.

Table 1. Example structure of a representative sample for model retraining

Message Type	Number of Examples	Feedback Reliability (%)	Sample Weight
Correctly Classified (Spam)	350	95	1.0
Correctly Classified (Legitimate)	420	94	1.0
False Positives	60	88	1.3
False Negatives	45	90	1.3
Low Confidence Messages	80	85	1.2

Source: compiled by the authors

The dataset was based on accumulated user feedback, taking into account not only the type of classification (correct or incorrect) but also the reliability of the provided feedback. As shown in Table 1, the structure of the dataset included five main categories: correctly classified spam and legitimate messages, two types of classification errors (false positives and false negatives), and messages for which the model exhibited a low confidence level in its decision. For each category, the average reliability of the feedback was also indicated, reflecting the proportion of confirmed or highly probable evaluations, as well as the weight within the dataset, which determined the influence of the respective subset on the model update process. At the final stage, the effectiveness of the SMS spam classification models was evaluated using five key metrics: Accuracy, Precision, Recall, F1-score, and ROC-AUC. The obtained values were summarised in an overall performance vector:

$$E = \{e_1, e_2, e_3, e_4, e_5\}, \quad (4)$$

where each component corresponded to a separate metric. For the interpretation of results, a linguistic scale for evaluating model performance was introduced:

$e \in (0.8; 1]$ – l_1 = "high performance", $e \in (0.6; 0.8]$ – l_2 = "above-average performance", $e \in (0.4; 0.6]$ – l_3 = "average performance", $e \in (0.2; 0.4]$ – l_4 = "low performance", $e \in [0; 0.2]$ – l_5 = "very low performance".

To achieve the aim of the study, a comprehensive approach was employed, combining theoretical and empirical methods. The research was conducted using an analytical method, reviewing the scientific literature devoted to SMS spam detection, including the use of ML algorithms, neural networks. At the initial stage, a thorough review of scientific literature was conducted, focusing on SMS spam detection methods, particularly those involving ML algorithms, neural networks, and hybrid architectures that integrate various text processing techniques. The proposed hybrid architecture of the SMS spam detection system was built on the principle of distributed processing, combining a local level (on the user's device) and a cloud level (centralised processing). At the local level, basic message classification was performed with minimal latency. Messages with uncertain or suspicious results were automatically forwarded to

the cloud level, where more complex and resource-intensive DL models (such as LSTM) were applied. A key element of the architecture was the classification result weighting mechanism, which aggregated the outcomes of the local and cloud models, taking into account: the confidence level of each model, the historical accuracy of the models on similar messages, the trustworthiness of the message source (where available).

Model updates were carried out asynchronously, based on aggregated suspicious messages received from a large number of users. To prevent training on anomalous or deliberately manipulated data (data poisoning), a knowledge update stabilisation mechanism was implemented in the cloud component of the system; specifically: a message buffer with controlled size and time window was used, the consistency of the statistical characteristics of new data was checked (for example average message length, frequency of spam tokens), mechanisms for anomaly filtering and weight-limited updates were applied to prevent overfitting to exceptional cases.

The model retraining mechanism was based on periodic analysis of the accumulated message buffer containing user feedback. If the number of unique new patterns exceeded threshold values, controlled retraining of the model was initiated, with the previous parameters retained as the baseline configuration. The model selection was guided by the need to achieve an optimal trade-off between classification speed, resource efficiency, and spam detection accuracy, considering the functional distribution of the system across both local and cloud environments. Such a hybrid architecture addresses the limitations of computing power on user devices while at the same time enabling the implementation of more complex algorithms within the cloud environment.

At the local level, Naïve Bayes and logistic regression models were employed, as they were characterised by high processing speed, low resource consumption, and ease of implementation. This ensures rapid decision-making without delays and independently of network connectivity. Logistic regression, although still a linear model, provides greater sensitivity to individual features (such as distinctive words or phrases), which in turn helped to improve the overall balance of classification. However, both local models are limited in their ability to detect complex or hidden patterns, as they do not account for sequential dependencies between words and do not support the modelling of non-linear relationships. To overcome these limitations, more powerful algorithms such as Random Forest and LSTM are employed at the cloud level. Random Forest, as an ensemble of decision trees, made it possible to model non-linear relationships between features and demonstrates resilience against overfitting, even when working with complex or noisy data. Its effectiveness lies in its ability to generalise information across multiple trees and filter out random anomalies. The deployment of this model in the cloud environment has enabled the processing of large volumes of data with flexible parameter tuning.

The highest performance indicators were achieved with the LSTM model, which belonged to the class of recurrent neural networks and is specialised in processing sequences. The LSTM architecture permitted to retain the context of preceding words, which was critically important for spam classification, as such messages often contain manipulative sequences, dynamic patterns, or veiled calls to action. Owing to the computational capabilities of the cloud environment, this model can be trained on the full dataset without simplifying its structure, thereby ensuring high sensitivity and strong generalisation ability. For the local level, Naïve Bayes and logistic regression models were chosen due to their high processing speed and moderate accuracy. These algorithms had low computational requirements, allowing for rapid classification without significant delays on user devices, even when processing power is limited. In addition, they performed well with vector representations of short texts, such as TF-IDF, which makes them effective for the initial filtering of messages.

At the cloud level, more sophisticated models were employed – Random Forest and LSTM. Random Forest provides flexibility and reliability, particularly when the feature set is limited, as its ensemble nature allows it to model complex non-linear relationships and respond robustly to noisy data. LSTM was selected as the primary model for deep analysis because it can capture the sequential order of words in a message, a capability crucial for uncovering hidden meanings in phishing texts. Owing to its architecture, LSTM can retain context and model dependencies in long sequences, significantly improving the classification accuracy of complex and dynamic spam patterns. In this way, the combination of fast and lightweight models at the local level with powerful deep learning algorithms in the cloud provided an optimal balance between processing speed, accuracy, and the system's adaptability to changes in the nature of SMS spam.

RESULTS AND DISCUSSION

The proposed SMS spam detection system featured a hybrid architecture, combining local and cloud levels of message processing. This approach enabled a balance between classification accuracy and data privacy (Al-Zebari *et al.*, 2025). Because of this loss function, the model optimised its weights during training, which reduced the number of misclassifications of messages as spam or legitimate content. Its high sensitivity to the probability distribution between classes provided flexibility when handling ambiguous or evolving message patterns. Within the proposed system, this contributed to more accurate detection of new or modified SMS spam patterns, even in a dynamic information environment where attackers continually adapt message content to bypass filters.

At the cloud level, the system aggregated suspicious messages received from a large number of devices. These data were used for dynamic knowledge updating through model retraining. The collection of such messages could be carried out with anonymisation and in accordance with privacy policies described in federated learning studies (Li *et al.*, 2024). The cloud model performed periodic retraining based on the updated dataset, which maintained the relevance of vector representations and prevented shifts in the statistical characteristics of the data over time, as is typical for SMS spam.

Dynamic updating was ensured through continuous learning mechanisms, allowing the model to adapt to new threats in real time (Boyko & Kovalchuk, 2023). Stability control techniques were applied, such as restricted parameter updates or the use of change buffers, which were introduced only after validation on test data. This approach prevented overfitting to anomalies and preserved the model's generalisation capability amid rapid changes in spam strategies. Retraining on anomalies can cause a model to memorise atypical or random patterns in the input data, reducing its ability to generalise and perform effectively on new examples. For instance, if a short burst of unusual but random messages is received and classified as spam, a model that updates without additional checks may adapt to these patterns and lose effectiveness in detecting more general forms of spam. Therefore, the use of change buffers and the validation of new parameters on a separate dataset helps determine whether the new data truly represents a sustained shift in patterns rather than a short-lived anomaly.

Preserving the model's generalisation capability was critically important in the context of the continual evolution of spam strategies. Spammers could periodically change keywords, message structure, or delivery methods, causing rapid shifts in the distribution of incoming data. The adaptive system needed to distinguish genuine emerging trends from random noise in the data. The application of update stabilisation methods prevented responses to isolated spikes, instead focusing adaptation on long-term patterns, thereby ensuring the robustness and relevance of the classification model in a dynamic environment. User or system feedback was also taken into account during the knowledge update process. This enabled the construction of an adaptive system capable of evolving in response to changes in the nature of SMS spam. Incorporating feedback supported not only the technical adaptation of models but also personalisation – the system could account for the individual perception of spam by different user groups. Similar architectures had already been examined in contemporary studies, for instance by S. Hossain *et al.* (2022) on multi-level spam filtering, confirming the effectiveness of multi-component systems with cyclical knowledge updating. Structurally, the operation of the proposed system was illustrated in Figure 1.

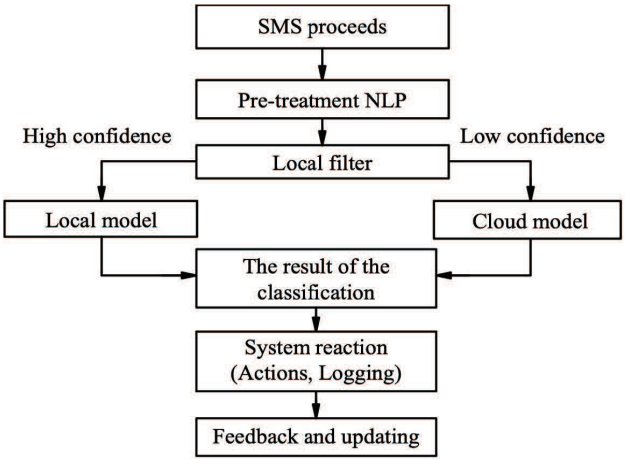

Figure 1. System operation flowchart: from message receipt to response
Source: compiled by the authors

The architecture of the proposed adaptive SMS spam detection system relied on the integration of local data processing with cloud technologies and supports the dynamic updating of knowledge models. The core process of handling incoming messages is implemented through a sequence of functional modules, each responsible for a specific task. The initial stage involved receiving an SMS message, which was then passed to a pre-processing module that utilises NLP techniques. That module was responsible for cleaning the text, normalising it, tokenising, and generating a vector representation of the message using approaches such as TF-IDF or word embeddings, as stated in the study by C.N. Mohammed & A.M. Ahmed (2024). In cases where the classifier demonstrated high confidence, the message was classified immediately at the local level. However, if the confidence level was insufficient, the message was redirected to the cloud-based model, which employed more advanced computational algorithms, such as BERT or LSTM, for more accurate classification as was proved in the research made by J. Prashob & S.Y. Yerima (2022). The classification results from both the local and cloud-based models were combined within the aggregated results module, which made the final decision regarding whether a message belongs to the spam category or to legitimate communications.

The next stage is the system's response, which involved carrying out the appropriate actions – filtering, notifying the user, or logging events. The final module was responsible for collecting feedback and updating the models through retraining, taking into account classification corrections based on feedback from users or administrators. The effectiveness of spam detection systems largely depended on their ability to adapt to new types of messages, which evolve over time. Within the proposed hybrid architecture, an adaptive knowledge update mechanism was introduced, ensuring the dynamic improvement of classification models based on user feedback. After classification, each message could

be evaluated by the user or system log as correctly or incorrectly classified. These responses were aggregated and stored in a dedicated feedback collection module. Subsequently, taking into account the quantity, type, and reliability of such feedback, a representative dataset was formed for retraining or further training of the models.

Assigning higher weights to incorrectly classified messages (1.3) and messages with low confidence (1.2) allowed the model to focus on the most critical cases that required revision of classification behaviour. Nevertheless, messages with high classification reliability and confirmed feedback were assigned a standard weight (1.0), ensuring stability in the training process and preventing overfitting on erroneous or noisy examples. This approach allowed a balance to be achieved between the model's adaptability to new types of messages and the stability of system behaviour in operational conditions, supporting the maintenance of high classification accuracy without the need for complete retraining of the model. To prevent shifts in the statistical characteristics of the data over time and avoid overfitting, quality filters were applied: messages with high classification confidence and confirmed feedback were assigned higher weights during the knowledge update process. Furthermore, updates were not instantaneous – a pre-aggregation approach was employed, which allowed the stability of the model to be controlled at the production level (Rojas-Galeano, 2021). This approach ensured the gradual adaptation of the system to changes in SMS spam patterns, maintaining high relevance and classification accuracy without the need for manual model review or complete retraining. Figure 2 presented a schematic representation of the system's dynamic update mechanism based on collected feedback:

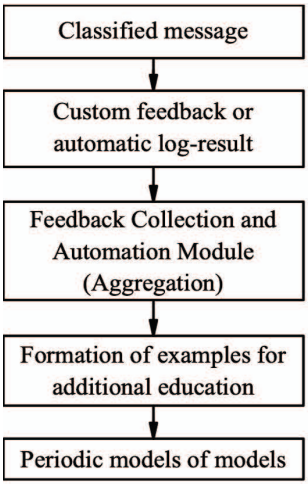

Figure 2. Adaptive knowledge update mechanism based on feedback
Source: compiled by the authors

The diagram illustrates the dynamic knowledge update workflow within the SMS spam detection system, based on user feedback. After each classification decision, users were given the opportunity to confirm or dispute the result. This feedback was collected and processed by evaluating source reliability, message recurrence, and other contextual features. An adaptive model was then constructed, which periodically updates both local and cloud-based components of the system. This feedback loop ensures that the model remains responsive to new spam behaviours, thereby mitigating the risk of performance degradation over time. To ensure an objective and comprehensive analysis of the effectiveness of the adaptive SMS spam detection system, a mechanism was developed that encompasses the complete cycle of constructing and evaluating classification models. At the initial stage, incoming SMS messages undergo pre-processing. This process includes text normalisation, which involves converting characters to lowercase and removing special characters, such as HTML tags or URLs. This was followed by tokenisation, where text had been split into individual words or tokens, and the removal of stop words that carry little semantic meaning. To further refine the textual data, stemming or lemmatisation techniques were applied to reduce words to their root forms. After these steps, the text was transformed into a numerical vector representation using approaches such as TF-IDF or modern embedding models.

To evaluate the performance of the adaptive SMS spam detection system, a set of quantitative metrics commonly applied in binary classification problems was employed. These metrics enable the assessment of not only the overall accuracy of the models but also their capability to identify spam messages while minimising false positive rates. Accuracy reflected the overall proportion of properly classified messages (both spam and legitimate) relative to the entire dataset. A high value of this metric indicates general consistency of the model; however, it may be insufficiently informative in cases of class imbalance, which is typical for spam filtering tasks. Precision characterised the proportion of truly spam messages among all messages classified as spam by the model. A high precision level indicates a low likelihood of false positives, which was critical in the context of avoiding the loss of legitimate messages. Recall, on the other hand, measured the model's ability to identify all existing spam messages within the dataset, including those that were more difficult to detect. This is particularly important in situations where missing harmful content was unacceptable from a security or user experience perspective.

The F1-score represented the harmonic mean of precision and recall, providing a balanced assessment of the model, which was particularly useful when both metrics are considered equally important for the task. A high F1-score indicated the presence of a classifier that was both accurate and sensitive. The Receiver Operating Characteristic – Area Under the Curve (ROC-AUC) was used to assess the model's effectiveness in differentiating between spam and legitimate messages

across varying classification thresholds. An AUC value approaching 1.0 indicates a strong ability to discriminate between classes. This metric was especially valuable in scenarios where accurate class separation was essential, as it encapsulates the trade-off between sensitivity and specificity by balancing false positive and false negative rates (Molina-Coronado *et al.*, 2023).

Precision and recall reflected opposite types of errors, and their balance is particularly important in cases of class imbalance. For example, when the "spam" class is rare, recall is more sensitive to how well the model detects this class, even if precision decreases as a result. Conversely, high precision combined with low recall indicates that the model detects only the "obvious" spam messages while ignoring more complex examples. The F1-score, as the harmonic mean of precision and recall, provides a more balanced assessment in situations of class imbalance. This was particularly useful when it is important to avoid both missing spam and misclassifying legitimate messages. ROC-AUC was invariant to class ratios, but in cases of strong imbalance it can sometimes be more useful to employ PR-AUC (Precision-Recall AUC), which was more sensitive to variations in the rare class. For example, when recall was high but precision was very low, this indicated that the model had been overly aggressive in labeling messages as spam, which could have resulted in the loss of important messages.

In such cases, even a high ROC-AUC did not guarantee the model's practical usefulness. Therefore, when analysing classification models in imbalanced-class scenarios, it was advisable to consider a set of metrics collectively and in light of the task's specific risk priorities. The comparative results of four different models, presented in Table 2, indicate a significant improvement in performance when using cloud-based algorithms employing ensemble and neural methods (Random Forest, LSTM) compared with classical statistical approaches (Naïve Bayes, Logistic Regression). The highest values for accuracy, F1-score, and ROC-AUC were achieved by the LSTM model, demonstrating the advantage of DL in tasks involving sequential text data processing. However, in tasks with a strong class imbalance – such as spam filtering, where legitimate messages predominate – accuracy can give a misleading impression of a model's effectiveness. Formally, if the "spam" class constitutes only 5% of all messages, even a "dumb" model that always predicts "not spam" would achieve an accuracy of approximately 95%, despite failing to detect any spam. Therefore, it is important to consider other metrics that more accurately reflect classification quality under such conditions.

Table 2. Metrics for evaluating the effectiveness of SMS spam classification models

Model	Accuracy	Precision	Recall	F1-measure	ROC-AUC
Naïve Bayes (local)	0.92	0.88	0.85	0.86	0.91
Logistic Regression	0.94	0.90	0.88	0.89	0.93
Random Forest (cloud)	0.97	0.95	0.93	0.94	0.97
LSTM (cloud)	0.98	0.96	0.95	0.95	0.98

Source: compiled by the authors

The results presented in Table 2 demonstrate a gradual increase in classification efficiency depending on the complexity of the model, its ability to generalise, and its ability to examine the context of incoming messages. The Naïve Bayes model, which assumed of conditional independence of features, demonstrated the lowest, though still acceptable, performance: accuracy was 0.92, precision 0.88, and recall 0.85. This level of effectiveness was typical for this model in text classification tasks, where the consideration of inter-word relationships was limited. The model was fast and straightforward to implement; however, it lacks the flexibility to adapt to more complex structures in the text, which partially explains the lower F1-score (0.86) and ROC-AUC (0.91) compared with other approaches. Naïve Bayes model, despite the assumption of conditional independence of features, demonstrates solid performance in short-text classification tasks when used with TF-IDF, achieving accuracy rates exceeding 96%.

Logistic Regression, which has a better capacity to model linear relationships between features, achieved higher performance: accuracy reached 0.94, precision 0.90, and recall 0.88. Despite the linearity of the model, it more effectively accounts for the weight of individual features (such as specific words or phrases characteristic of spam), allowing for a balanced F1-score (0.89) and AUC (0.93). However, the inability to model complex non-linear relationships and sequences somewhat limited its overall performance. Random Forest, as a representative of ensemble tree-based methods, demonstrated substantially higher performance: accuracy was 0.97, precision 0.95, recall 0.93, F1-score 0.94, and AUC 0.97. This is attributable to the model's ability to generalise even in cases of complex or noisy data. The use of multiple independent trees reduces the risk of overfitting and better captures the variability of spam patterns. In this experiment, Random Forest was deployed in a cloud environment, which allowed for the processing of larger data volumes and the utilisation of optimised parameters.

The highest performance was demonstrated by the LSTM model – a neural network specialised in sequence processing. Its accuracy reached 0.98, precision – 0.96, recall – 0.95, F1-score – 0.95, while the ROC-AUC achieved 0.98. Owing to its architecture, which is capable of retaining context at the word-sequence level,

LSTM proved most effective in recognising hidden patterns within spam messages, particularly those involving manipulative vocabulary, sequential instructions, or "technical" phrases (e.g., "click here", "you have won"). Furthermore, the cloud environment provided sufficient computational resources to train the deep model on the full dataset without compromising its complexity. The overall trend in performance indicates that the transition from simple statistical methods to DL approaches provided a substantial improvement in results, particularly in tasks with high semantic complexity and intricate contextual dependencies, which are characteristic of modern spam. Thus, all the models considered demonstrated high effectiveness (l_1) in the task of SMS spam classification; however, the most successful was the LSTM neural network operating within a cloud environment.

M. Ahmadi *et al*. (2025) investigated the effectiveness of various feature extraction methods and classification algorithms for detecting SMS spam. They compared six classical and deep classifiers – Naïve Bayes, K-Nearest Neighbours, Support Vector Machines, Linear Discriminant Analysis, Decision Trees, and Deep Neural Networks – in combination with two text vectorisation methods: bag-of-words and TF-IDF. The study results showed that the TF-IDF method significantly outperformed bag-of-words across all classifiers, and the combination of TF-IDF with Naïve Bayes achieved the highest accuracy of 96.2%. Compared with other models, such as Support Vector Machines and Deep Neural Networks, Naïve Bayes with TF-IDF demonstrated superior accuracy and a better balance between detecting spam and legitimate messages.

These findings underscored the importance of selecting an appropriate feature extraction method to enhance the effectiveness of a spam detection system. Compared with the study, which also employs Naïve Bayes at the local level with TF-IDF for fast and accurate initial classification, the results of this research support the choice of lightweight and fast models for preliminary filtering. However, unlike their approach, this hybrid method complements the local classification with more sophisticated cloud-based models, thereby improving the system's accuracy and adaptability to new spam patterns. At the cloud level, suspicious messages from multiple devices were aggregated while adhering to privacy and anonymisation requirements (federated learning). This enabled the formation of updated training datasets and prevents changes in the statistical properties of the data over time, which could otherwise lead to model performance degradation. The use of continuous learning mechanisms and model stability control techniques (such as change buffers and validation set checks) helped avoid overfitting to short-term anomalies and maintains the system's generalisation capability over the long term.

D. Honeycutt *et al*. (2020) investigated the impact of interactive user feedback in human-in-the-loop machine learning systems. The authors conducted an experiment using a simulated image object recognition system to assess how the ability to provide feedback affects users' trust in the system and their perception of model accuracy. Although interactive feedback could potentially improve system performance, the results showed a decrease in both trust and perceived accuracy among users who participated in providing feedback. This suggested that the mere opportunity for users to intervene in the system's operation can have unexpected psychological effects, influencing technology acceptance. Comparing these results with the present study, where integration of user feedback and system logs was used to update models and personalise spam filtering, it is important to consider potential challenges in maintaining user trust. Although the proposed approach enhances system adaptability and its effectiveness in detecting new threats, it highlighted the importance of balancing automation with user involvement to avoid undermining trust or diminishing the user experience. Therefore, improving the models should take into account not only technical aspects but also interface design and interaction methods with end users.

Experimental results on the publicly available SMS Spam Collection Dataset confirmed the high accuracy and adaptability of the proposed system. The use of a comprehensive sampling structure for retraining, which accounts for the weights of different message types and the reliability of feedback, enabled an effective balance between model stability and flexibility. In particular, increased focus on misclassified cases and messages with low confidence helped reduce errors and improved overall classification quality without the risk of overfitting to random noise in the data. The conclusions align with the findings of the aforementioned studies: hybrid architectures indeed enhance both accuracy and adaptability, while the incorporation of privacy and robustness mechanisms represents a contemporary trend. On the other hand, the approach differed by integrating stability control mechanisms, weighted sampling, and a feedback loop, which collectively increased the reliability of the system in real-time operation.

All in all, the proposed hybrid SMS spam detection system reflects the current global trend towards developing adaptive, scalable, and personalised solutions for real-time information security. It successfully combines the advantages of local processing with cloud technologies, providing high accuracy, stability, and the ability to respond rapidly to emerging threats in a dynamic information environment. An important advantage of the proposed architecture was that cloud-level models are activated only when the local model exhibits low confidence or when new suspicious patterns typical of spam are detected. This conditional delegation strategy provided the ability to optimise the use of computational resources and reduce server load, while at the same time maintaining high classification accuracy. Thus, the selected set of models implemented the

principle of functional distribution of computations: simple and fast algorithms provide preliminary filtering on the user's device, while more complex models in the cloud performed deeper analysis only when required. This approach combined the advantages of speed, adaptability, and scalability, ensuring the reliable operation of the system in real time under the conditions of a dynamically changing spam environment.

CONCLUSIONS

The conducted study demonstrated the effectiveness of a hybrid SMS spam detection system that combines lightweight, local models for rapid filtering with deep learning solutions deployed in the cloud. This architecture ensures high classification accuracy while minimising computational costs, making it particularly suitable for devices with limited processing power. The integration of fast local evaluation with more resource-intensive cloud-based analysis reduces latency and optimises overall performance. Dynamic updating of knowledge models, informed by both user feedback and system logs, enables the system to adapt promptly to emerging spam patterns. This adaptability ensures robust generalisation, even as the statistical properties of the data evolve over time.

The system's performance was evaluated using the publicly available SMS Spam Collection Dataset and standard classification metrics, including accuracy, precision, recall, F1-score, and ROC-AUC. Comparative analysis showed that deep learning models – particularly LSTM – significantly outperformed traditional classifiers across all metrics, achieving an accuracy and AUC of 0.98. These results highlight the model's strong ability to distinguish between spam and legitimate content, even in complex scenarios. Among the tested methods (Naïve Bayes, Logistic Regression, Random Forest), LSTM achieved the highest performance and was utilised at the cloud level of the system. While classical models such as Naïve Bayes demonstrated acceptable results (accuracy of 0.92 and F1-score of 0.86), their inability to capture sequential dependencies between words limited their effectiveness in detecting more sophisticated or hidden patterns. This finding underscores the necessity of leveraging advanced neural architectures within cloud-based components.

Future research should focus on enhancing adaptive learning mechanisms through the development of more flexible feedback processing strategies, improving accuracy with limited data, and reducing update latency in the cloud. Moreover, the ethical use of personal data and the protection of user privacy during large-scale SMS processing in cloud environments remain critical areas for ongoing investigation.

ACKNOWLEDGEMENTS

None.

FUNDING

None.

CONFLICT OF INTEREST

None.

REFERENCES

[1] Abid, M.A., Ullah, S., Siddique, M.A., Siddique, M.A., Mushtaq, M.F., Alijedaani, W., & Rustam, F. (2022). Spam SMS filtering based on text features and supervised machine learning techniques. *Multimedia Tools and Applications*, 81, 39853-39871. doi: 10.1007/s11042-022-12991-0.

[2] Ahmadi, M., Khajavi, M., Varmaghani, A., Ala, A., Danesh, K., & Javaheri, D. (2025). Leveraging large language models for cybersecurity: Enhancing SMS spam detection with robust and context-aware text classification. *ArXiv*. doi: 10.48550/arXiv.2502.11014.

[3] Al Maruf, A., Al Numan, A., Haque, M.M., Jidney, T.T., & Aung, Z. (2023). Ensemble approach to classify spam SMS from Bengali text. In M. Singh, V. Tyagi, P. Gupta, J. Flusser & T. Ören (Eds.), *Advances in computing and data sciences. ICACDS 2023* (pp. 440-453). Cham: Springer. doi: 10.1007/978-3-031-37940-6_36.

[4] Almeida, T. & Hidalgo, J. (2011). SMS spam collection. *UCI Machine Learning Repository*. doi: 10.24432/C5CC84.

[5] Al-Zebari, A., Barwary, M., Omar, N., Zebari, N.A., & Zebari, D.A. (2025). Deep learning hybrid approach for accurate SMS spam identification. *Journal of Information Systems Engineering and Management*, 10(10s). doi: 10.52783/jisem.v10i10s.1426.

[6] Baaqeel, H., & Zagrouba, R. (2020). Hybrid SMS spam filtering system using machine learning techniques. In *2020 21st international Arab conference on information technology (ACIT)* (pp. 1-8). Giza: IEEE. doi: 10.1109/ACIT50332.2020.9300071.

[7] Bäckman, D. (2019). *Evaluation of machine learning algorithms for SMS spam filtering*. (Bachelor's thesis, Umeå University, Umeå, Switzerland).

[8] Boyko, N., & Kovalchuk, R. (2023). Data update algorithms in the machine learning system. *Computer Systems and Information Technologies*, 1, 6-13. doi: 10.31891/csit-2023-1-1.

[9] Gadde, S., Lakshmanarao, A., & Satyanarayana, S. (2021). SMS spam detection using machine learning and deep learning techniques. In *2021 7th international conference on advanced computing and communication systems (ICACCS)* (pp. 358-362). Coimbatore: IEEE. doi: 10.1109/ICACCS51430.2021.9441783.

[10] Gomaa, W.H. (2020). The impact of deep learning techniques on SMS spam filtering. *International Journal of Advanced Computer Science and Applications*, 11(1), 544-549. doi: 10.14569/IJACSA.2020.0110167.

[11] Honeycutt, D.R., Nourani, M., & Ragan, E.D. (2020). Soliciting human-in-the-loop user feedback for interactive machine learning reduces user trust and impressions of model accuracy. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, 8(1), 63-72. doi: 10.1609/hcomp.v8i1.7464.

[12] Hossain, S.M.M., Sumon, J.A., Sen, A., Alam, M.I., Kamal, K.M.A., Alqahtani, H., & Sarker, I.H. (2022). Spam filtering of mobile SMS using CNN-LSTM based deep learning model. In *Hybrid intelligent systems* (pp. 106-116). Cham: Springer. doi: 10.1007/978-3-030-96305-7_10.

[13] Kalyani, V.V., Rama Sundari, M.V., Neelima, S., Satya Prasad, P.S., PattabhıRama Mohan, P., & Lakshmanarao, A. (2024). SMS spam detection using NLP and deep learning recurrent neural network variants. In *2024 international conference on cognitive robotics and intelligent systems (ICC – ROBINS)* (pp. 92-96). Coimbatore: IEEE. doi: 10.1109/ICC-ROBINS60238.2024.10533895.

[14] Li, Y., Zhang, R., Rong, W., & Mi, X. (2024). SpamDam: Towards privacy-preserving and adversary-resistant SMS spam detection. *ArXiv*. doi: 10.48550/arXiv.2404.09481.

[15] Mohammed, C.N., & Ahmed, A.M. (2024). A semantic-based model with a hybrid feature engineering process for accurate spam detection. *Journal of Electrical Systems and Information Technology*, 11, article number 26. doi: 10.1186/s43067-024-00151-3.

[16] Molina-Coronado, B., Mori, U., Mendiburu, A., & Miguel-Alonso, J. (2023). Efficient concept drift handling for batch Android malware detection models. *ArXiv*. doi: 10.48550/arXiv.2309.09807.

[17] Oyeyemi, D.A., & Ojo, A.K. (2024). SMS spam detection and classification to combat abuse in telephone networks using natural language processing. *Journal of Advances in Mathematics and Computer Science*, 38(10), 144-156. doi: 10.9734/jamcs/2023/v38i101832.

[18] Prashob, J., & Yerima, S.Y. (2022). A comparative study of word embedding techniques for SMS spam detection. In *14th IEEE international conference on computational intelligence and communication networks (CICN 2022)* (pp. 149-155). Al-Khobar: IEEE. doi: 10.1109/CICN56167.2022.10008245.

[19] Rojas-Galeano, S. (2021). Using BERT encoding to tackle the Mad-lib attack in SMS spam detection. *ArXiv*. doi: 10.48550/arXiv.2107.06400.

[20] 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. doi: 10.1109/ACCESS.2024.3364671.

Vats, S., Shastri, S., & Mehta, S. (2024). Federated learning for SMS spam detection: A privacy-focused approach. *2024 15th international conference on computing communication and networking technologies (ICCCNT)* (pp. 1-5). Kamand: IEEE. doi: 10.1109/ICCCNT61001.2024.10724879.

Адаптивна гібридна система виявлення SMS-спаму з самонавчанням за фідбеком користувача

Ігор Лях

Доктор технічних наук, професор
Державний вищий навчальний заклад «Ужгородський національний університет»
88000, пл. Народна, 3, м. Ужгород, Україна
https://orcid.org/0000-0001-5417-9403

Андрій Чорній

Кандидат технічних наук, доцент
Черкаський державний технологічний університет
18006, б-р Шевченка, 460, м. Черкаси, Україна
https://orcid.org/0000-0002-0989-711

Оксана Лутак

Магістрантка
Державний вищий навчальний заклад «Ужгородський національний університет»
88000, пл. Народна, 3, м. Ужгород, Україна
https://orcid.org/0009-0002-0798-6928

Мар'ян Ценкнер

Магістрант
Державний вищий навчальний заклад «Ужгородський національний університет»
88000, пл. Народна, 3, м. Ужгород, Україна
https://orcid.org/0009-0003-2752-7398

Анотація. У статті представлено комплексний підхід до виявлення SMS-спаму на основі гібридної архітектури, яка поєднує локальні алгоритми обробки повідомлень із високопродуктивними хмарними моделями глибокого навчання. Такий підхід дозволяє досягти балансу між точністю та конфіденційністю обробки вхідних повідомлень. Метою дослідження було створення інтелектуальної гібридної системи виявлення SMS-спаму, яка забезпечувала б високу точність класифікації, підтримку актуальності знань, персоналізацію для користувачів та здатність адаптуватися до нових шаблонів атак. Для досягнення мети дослідження було застосовано комплексний аналітичний підхід, що поєднував детальний огляд наукової літератури з питань виявлення спаму в SMS-повідомленнях, включаючи машинне навчання, нейронні мережі та гібридні методи, з емпіричним аналізом. Для реалізації класичних моделей машинного навчання (Naïve Bayes, Logistic Regression, Random Forest) використовувалися стандартні бібліотеки машинного навчання, а для глибокого навчання – фреймворки, що підтримують рекурентні нейронні мережі, зокрема Long Short-Term Memory та трансформерні архітектури. Тестування системи на відкритому датасеті SMS Spam Collection з використанням метрик Accuracy (до 0,98), F1-score (до 0,95) та ROC-AUC (до 0,98). Додатково було реалізовано механізм динамічного оновлення знань через зворотний зв'язок користувача та запропоновано вагову систему оцінки достовірності фідбеку. У ході дослідження було розроблено багаторівневу систему, що виконувала початкову класифікацію на пристрої користувача з можливістю делегування обробки хмарному модулю у випадках невизначеності. У порівнянні з базовими підходами, гібридна архітектура продемонструвала покращення точності класифікації, зниження кількості хибнопозитивних і хибнонегативних спрацьовувань, а також підвищену адаптивність до змін у структурі spam-повідомлень. Агрегація підозрілих повідомлень у хмарі забезпечувала ефективне донавчання моделей у випадках концептуального зсуву. Практична цінність результатів полягає в можливості інтеграції розробленої системи для мобільних платформ, а також у корпоративні засоби інформаційної безпеки з метою фільтрації SMS-контенту та захисту кінцевих користувачів від соціальної інженерії

Ключові слова: обробка природної мови; довга короткочасна пам'ять; архітектура спаму; повідомлення; метрика