IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 15, No. 1, February 2026, pp. 909~918
ISSN: 2252-8938, DOI: 10.11591/ijai.v15.i1.pp909-918

Deep learning-based spam detection for WhatsApp chatbot fallback reduction

Satrio Sadewo, Amalia Zahra
Department of Computer Science, BINUS Graduate Program-Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia

Article Info

Article history:

Received Sep 22, 2025
Revised Jan 2, 2026
Accepted Jan 22, 2026

Keywords:

Deep learning
Fallback response
Machine learning
Natural language processing
Spam detection
Transformer models
WhatsApp chatbot

ABSTRACT

Chatbots on WhatsApp are widely used for customer service, but their effectiveness is often undermined by fallback responses when user input cannot be understood. A major cause of these fallbacks is unsolicited spam, which disrupts interactions and reduces service quality. This study develops and evaluates a spam detection system aimed at reducing fallback rates and enhancing user experience. A comparative analysis was conducted between traditional machine learning models (support vector machine (SVM) and decision tree (DT)) and advanced deep learning architectures, including long short-term memory (LSTM) variants (vanilla, bidirectional, stacked, convolutional neural network (CNN)-LSTM, and encoder-decoder) and transformer-based models (bidirectional encoder representations from transformers (BERT)-base, DistilBERT, and cross-lingual language model-robustly optimized BERT pretraining approach (XLM-ROBERTa)). Using 170,000 messages sampled from 18 million interactions collected between July 2022 and December 2023, the models were assessed with standard evaluation metrics. Results show that CNN-LSTM and DistilBERT achieved the most robust performance. CNN-LSTM attained a precision of 0.92, recall of 0.91, F1-score of 0.91, and accuracy of 0.94, while DistilBERT achieved precision of 0.92, recall of 0.89, F1-score of 0.90, and accuracy of 0.93. These findings highlight their superior ability to capture contextual patterns in spam messages. Implementing such models is expected to significantly lower fallback rates, thereby improving chatbot reliability and user satisfaction.

Corresponding Author:

Satrio Sadewo
Department of Computer Science, BINUS Graduate Program-Master of Computer Science
Bina Nusantara University
Jakarta, Indonesia
Email: satrio.sadewo@binus.ac.id

1. INTRODUCTION

In the contemporary digital landscape, WhatsApp has established itself as a critical communication medium, serving a global user base exceeding two billion [1]. The integration of chatbots onto this platform has precipitated a paradigm shift in customer service, marketing strategies, and user engagement protocols [2]. Although chatbots present considerable advantages, contributing to an average revenue increase of 67% and accounting for 26% of total sales interactions [3], their operational efficacy is frequently impeded by technical limitations [4]. A predominant issue is the high rate of "fallback" responses, wherein the chatbot is unable to interpret a user's query, consequently delivering a generic and unhelpful reply. This phenomenon not only diminishes the user experience but also signifies underlying architectural or training inefficiencies [5]. A principal factor contributing to the high rate of fallback

responses is the pervasiveness of unsolicited spam messages. Such messages disrupt the intended conversational trajectory, trigger non-productive fallback responses, and may introduce security vulnerabilities, including phishing attacks [6], [7]. This research confronts the challenge of spam-induced fallback responses within a WhatsApp chatbot ecosystem. The investigation leverages a substantial dataset of 18 million messages, collected from an operational chatbot service, of which 6 million (33%) culminated in a fallback. The central objective is to engineer a robust spam detection model capable of reducing the fallback rate to a target threshold of 15% or less.

Prior research has affirmed the utility of machine learning algorithms such as decision trees (DT) and support vector machine (SVM) for spam detection across various platforms, including email and SMS [8], [9]. More recently, deep learning models, notably long short-term memory (LSTM) and bidirectional encoder representations from transformers (BERT), have demonstrated superior performance in managing the complexity and contextual nuances of natural language, attaining high accuracy in spam classification tasks [10], [11]. To provide a clear benchmark of the current state-of-the-art and highlight these comparative findings, Table 1 presents a comparative taxonomy of recent spam detection studies across various platforms and methodologies.

This study addresses a critical gap by evaluating spam detection models specifically on WhatsApp chatbot data, which presents a unique and non-trivial challenge compared to previously studied domains like email and SMS. Unlike traditional spam, WhatsApp spam is characterized by extreme brevity, heavy use of informal language and slang, and contextual mimicry that often imitates legitimate user queries to evade detection [12]. These characteristics render many frequency-based machine learning features less effective and necessitate advanced deep learning models capable of understanding nuanced, context-dependent patterns. The primary contribution of this work is a rigorous, direct comparison of traditional and advanced deep learning models on a large-scale, real-world dataset to identify the most robust architecture for this specific, challenging environment. To provide a clear structure for this investigation, this paper aims to answer two primary research questions: first, (RQ1) how can spam messages that cause fallbacks in a WhatsApp chatbot be effectively detected using traditional machine learning (SVM and DT) versus deep learning models (LSTM variants, and BERT variants)?; and second, (RQ2) to what extent do deep learning models outperform traditional machine learning models in minimizing the fallback rate through superior spam classification on this unique dataset?

Table 1. Comparative taxonomy of state-of-the-art spam detection models

Title/author(s)	Platform	Method(s)	Evaluation metrics	Key result(s)
Email spam detection using deep learning approach [6]	Email	LSTM, BiLSTM, BERT	Accuracy, precision, recall, F1-score	BERT: 99.14% (highest accuracy)
SMS spam detection using machine learning and deep learning techniques [10]	SMS	naive Bayes, LR, RF, SVM, KNN, DT, LSTM	Precision, recall, accuracy	LSTM: 98.5% (highest accuracy)
E-mail spam detection using machine learning [12]	Email	DT, RF, naive Bayes, SVM, LR, MLP	Accuracy	MLP: 98% (highest accuracy)
Email spam classification using DistilBERT [13]	Email	DistilBERT	Accuracy	DistilBERT: 97.84% (validation accuracy)
LSTM networks for email spam classification [14]	Email	LSTM	Accuracy	LSTM: 97.4% accuracy
A comprehensive review on email spam classification using machine learning algorithms [9]	Email	SVM, naive Bayes, DT, RF, neural networks	Accuracy, precision, recall	SVM: 98.32% (highest accuracy)
SMS spam classification using machine learning techniques [8]	SMS	naive Bayes, LR, SVM, RF	Accuracy	SVM: 98.79% (highest accuracy)

2. METHOD

The research methodology was systematically designed to facilitate the development and subsequent evaluation of a spam detection model. As detailed in the following sections, this process began with data collection and labeling, followed by preprocessing, model building, hyperparameter tuning, and finally, a comprehensive model evaluation. The procedural framework, depicted in Figure 1, commenced with data acquisition and concluded with this comprehensive evaluation of the resultant models. Figure 1 illustrating the sequential steps from data collection and annotation to model evaluation.

2.1. Data collection and preparation

The dataset was procured from the interaction logs of a WhatsApp chatbot, covering the period from July 2022 to December 2023 and encompassing 18 million messages. For the purpose of this study, a

representative sample of 170,000 messages that had triggered a fallback response was isolated. This dataset was subsequently partitioned into training (70%), validation (15%), and testing (15%) subsets [15]. The partitioning was performed chronologically to rigorously assess the model's capacity for temporal generalization.

A rigorous annotation process was then conducted by a panel of three domain experts (composed of one data scientist and two product owner representatives) to classify each message as either 'spam' or 'non-spam' [16]. This process was governed by a strict set of classification rules defined in collaboration with the product owner to ensure domain relevance, as outlined in Table 2. To ensure consistency and minimize subjective bias, an initial validation was performed where all three experts independently labeled a sample of 15,000 messages; the final label for this set was determined by a majority vote. Following this manual validation, a semi-supervised approach was used to annotate the full dataset: a preliminary model trained on the 15,000-message set was used to provide initial predictions, which were then validated and corrected by the expert panel to ensure high-quality labels for the final training data. This labeling process resulted in the data distribution across the training, validation, and testing subsets detailed in Table 3, which notably includes temporal variation in spam prevalence.

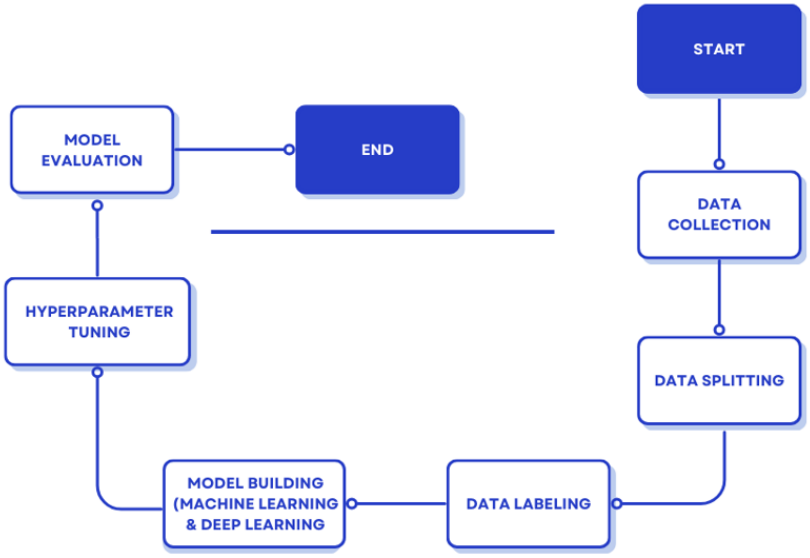

Figure 1. Research methodology flowchart

Table 2. Spam classification criteria in data labeling

Spam rule	Example message
Irrelevant messages	"She was crying in the lakes", "*Jual beli tanah lengkap di sini* "
Promotional messages (non-company)	"*Situs pertaruhan online, sini kakak join*", "*Saya menjual lampu baca, bisa ke link ini ya*"
Messages with only a single character (unless specific meaning)	"A B C D E F G Z,?/'; " (exception: Y/N)
Messages with meaningless character combinations	"BNVMMDFKDsfsd", "Dsadadkheur34324235"
Profane or vulgar words without context	(Specific vulgar examples)
Copied/pasted chat (except report formats)	"*Maaf, pastikan tidak ada penulisan dalam chat*", "*Apa bisa saya bantu??*"
Messages with links (except company domains)	"www.instagram.com/akujualnasi", "www.hotspot.com"

Table 3. Label distribution per data subset

Dataset name	Label		Total
	Non-spam	Spam	
Data training 072022 to 072023	48,020	16,980	65,000
Data validation 082023	15,153	9,831	24,984
Data testing 092023	14,923	10,058	24,981
Data testing 102023	16,971	8,027	24,998
Data testing 112023	12,582	2,415	14,997
Data testing 122023	12,359	2,641	15,000

2.2. Data preprocessing

Prior to model training, the textual data was subjected to a comprehensive preprocessing pipeline to ensure its cleanliness and normalization. A critical step for improving model performance [17]. This pipeline involved several stages: conversion of all text to lowercase; removal of punctuation and special characters; normalization of slang and informal terminology to their standard lexical forms; elimination of stopwords [18]; tokenization to segment the text into discrete words [19]; and the application of stemming to reduce words to their morphological roots [20].

2.3. Feature extraction and modeling

For the conventional machine learning models (DT and SVM), the preprocessed text data was transformed into numerical feature vectors. Various feature extraction techniques were evaluated, including CountVectorizer, HashingVectorizer, term frequency-inverse document frequency (TF-IDF), Word2Vec, and FastText. Following preliminary evaluations, TF-IDF was determined to be the optimal method for feature representation, consistent with its widespread successful application in text classification literature [21].

For the deep learning paradigm, two principal architectural categories were investigated:

i) LSTM: a range of LSTM variants were implemented and tested, including vanilla LSTM, stacked LSTM, bidirectional LSTM, and a hybrid CNN-LSTM architecture. These models are recognized for their proficiency in capturing sequential dependencies within textual data [22]–[24].

ii) BERT: pre-trained BERT models, specifically BERT-base, DistilBERT, and cross-lingual language model-robustly optimized BERT pretraining approach (XLM-ROBERTa), were fine-tuned for the binary spam classification task. The efficacy of these models is derived from their advanced ability to process the bidirectional context of words within a sentence [11], [13], [25].

To provide the requested methodological clarity, Table 4 details the architectures for the two top-performing models. The CNN-LSTM architecture was built from scratch, while the DistilBERT architecture shows the classification head added on top of the pre-trained base model for fine-tuning. The models were implemented utilizing the TensorFlow and Keras frameworks. A critical phase of the methodology was hyperparameter tuning, wherein techniques such as grid search were employed to systematically identify the optimal configuration of each model's parameters (e.g., learning rate, number of layers, and dropout rate) to maximize predictive performance [26]. The final optimal hyperparameters for the top-performing models were identified from this process. For the SVM model, a polynomial kernel with a degree of 2 and a C value of 1 was used. For the DistilBERT model, the fine-tuned architecture consisted of three hidden dense layers with 512, 256, and 128 neurons, respectively, each followed by a dropout layer with rates of 0.5, 0.4, and 0.3. An Adam optimizer with a learning rate of 0.0001 was utilized for training.

Table 4. Architectural schematic of CNN-LSTM and DistilBERT (fine-tuning)

Layer	Architecture of CNN-LSTM	Architecture of fine-tuning DistilBERT
Input	Embedding Layer	Input from DistilBERT Base (768 unit)
1	Conv1D (64 filter, kernel=5)	Dense (512 unit)
2	MaxPooling1D (pool size=2)	Dropout (Rate=0.5)
3	LSTM (64 unit)	Dense (256 unit)
4	-	Dropout (Rate=0.4)
5	-	Dense (128 unit)
6	-	Dropout (Rate=0.3)
Output	Dense (1 unit, 'sigmoid')	Dense (1 unit, 'sigmoid')

3. RESULTS AND DISCUSSION

The performance of the trained models was rigorously evaluated using precision, recall, F1-score, and accuracy [27]. This evaluation was conducted on distinct test datasets from four consecutive months (September to December 2023) to not only measure performance but also to assess the models' stability and robustness against temporal shifts in data patterns. This longitudinal approach is critical, as the nature of spam can evolve over time [28]. All results presented in the summary tables reflect the average performance across these four testing periods unless otherwise specified.

3.1. Preliminary feature extraction evaluation

Based on the preliminary results shown in Table 5, TF-IDF was selected as the optimal vectorization technique for the final machine learning model comparisons. While HashingVectorizer and CountVectorizer gave the DT model high accuracy (0.92), their performance with SVM was significantly lower, with recall scores of 0.73 and 0.68, respectively. Therefore, TF-IDF combined with SVM, which yielded a strong and more balanced performance (Accuracy 0.88, F1-score 0.87), was identified as the most robust choice for both algorithms.

Table 5. Summary of evaluation results of methods with feature extraction

Feature extraction	Method	Precision	Recall	F1-score	Accuracy
CountVectorizer	DT	0.91	0.92	0.92	0.92
	SVM	0.85	0.68	0.67	0.74
Word2Vec	DT	0.87	0.87	0.87	0.87
	SVM	0.85	0.86	0.85	0.85
FastText	DT	0.88	0.84	0.85	0.86
	SVM	0.89	0.85	0.87	0.88
HashingVector	DT	0.92	0.93	0.92	0.92
	SVM	0.86	0.73	0.74	0.78
TfidfVectorizer	DT	0.91	0.92	0.92	0.92
	SVM	0.89	0.87	0.87	0.88

3.2. Model performance

Following an extensive hyperparameter tuning process, both the traditional machine learning and advanced deep learning models demonstrated strong predictive capabilities. However, a distinct performance hierarchy was observed, with the deep learning models consistently outperforming their traditional counterparts. This advantage is attributable to their inherent capacity to learn complex, hierarchical features and understand contextual relationships within the text, which is a limitation for algorithms like DT and SVM that rely on more superficial, frequency-based features. Table 6 presents the average performance metrics of the most notable models over the four-month testing period.

Table 6. Average performance comparison of all models across the four-month test period (September to December 2023)

Models	Precision	Recall	F1-score	Accuracy
DT	0.90	0.92	0.91	0.93
SVM	0.90	0.92	0.91	0.93
Stacked LSTM	0.91	0.91	0.91	0.94
CNN-LSTM	0.92	0.91	0.91	0.94
Encoder decoder LSTM	0.91	0.90	0.91	0.94
BERT-BASE	0.91	0.89	0.90	0.92
DistilBERT	0.92	0.89	0.90	0.93
XLM-Roberta	0.92	0.79	0.83	0.89

The empirical results identified the CNN-LSTM and DistilBERT models as the preeminent architectures in terms of performance. The hybrid CNN-LSTM model achieved the highest overall accuracy (0.94), showcasing the power of its architecture. This synergy is particularly effective for identifying spam; the CNN layers act as a powerful feature extractor, identifying local n-gram patterns and keywords often associated with spam (e.g., *'promo'* or *'link ini'*), while the LSTM layers analyze the sequence in which these patterns appear, capturing the contextual structure of the message.

DistilBERT, a lighter and faster version of BERT, also demonstrated excellent, well-rounded performance with high precision (0.92) and accuracy (0.93). Its strength lies in its ability to retain a powerful grasp of bidirectional context, even as a distilled model. This is crucial for distinguishing sophisticated spam that mimics conversation. For example, in a message like "*Kakak dm*", a unidirectional model might be confused, but a bidirectional model can analyze the context from both directions to better assess the intent, relating it to patterns seen in other promotional spam.

In contrast, the multilingual model XLM-ROBERTa, while achieving high precision, exhibited a notable deficiency in recall (0.79). This suggests that its generalized, cross-lingual training was less effective for this domain-specific task. The model, trained on 100 languages, may lack the specific tokenization and embedding representations needed to understand the nuances of Indonesian slang, informalities, and context-specific spam (e.g., "*Situs pertaruhan online*") as effectively as the models fine-tuned specifically on this dataset.

3.3. Analysis of misclassifications

A more granular analysis of the misclassifications highlights the persistent challenges in automated chatbot spam detection. Figures 2 and 3 present the confusion matrices for the top-performing CNN-LSTM and DistilBERT models, respectively. These figures detail their performance across the four-month testing period.

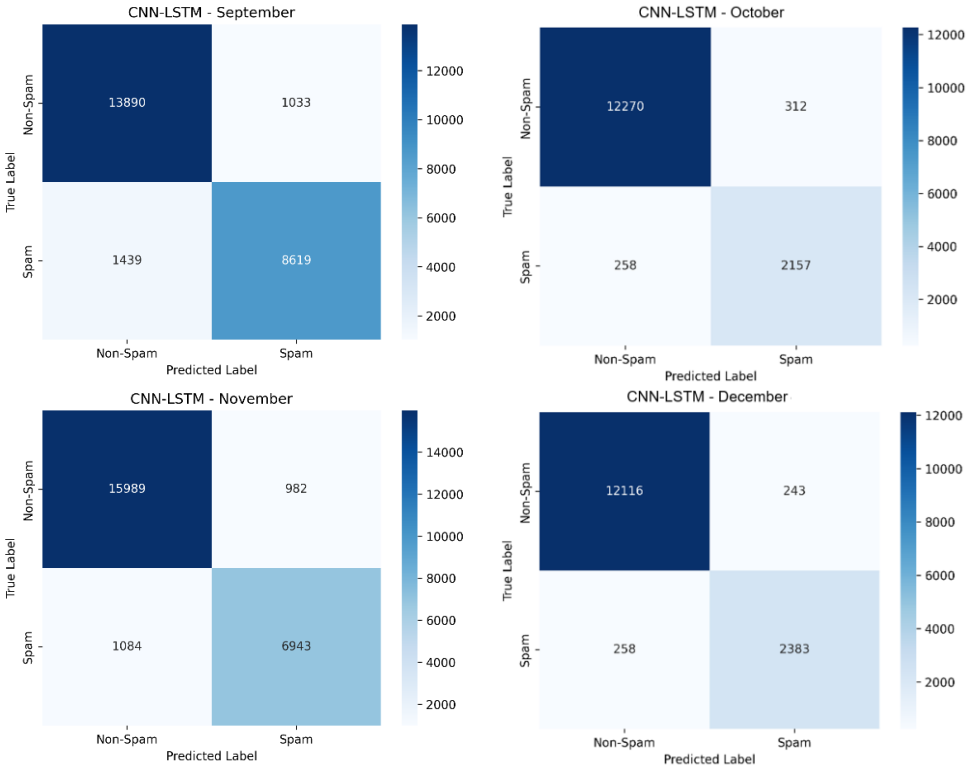

Figure 2. Monthly confusion matrices for the CNN-LSTM model, showing predictive performance on the test sets from September to December 2023

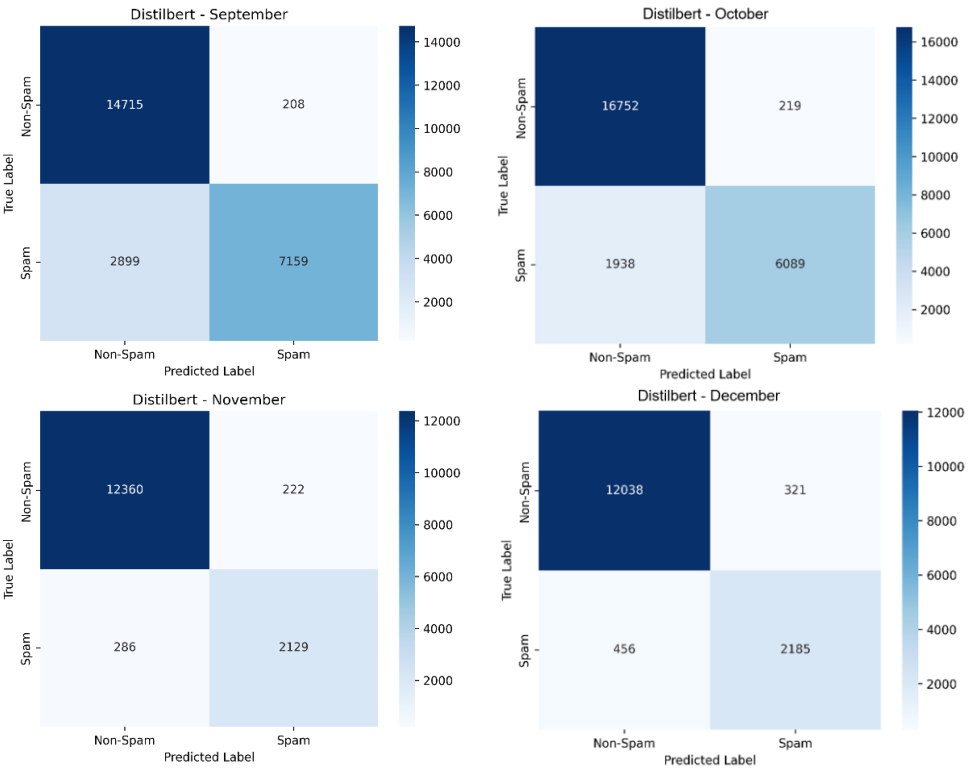

Figure 3. Monthly confusion matrices for the DistilBERT model, showing predictive performance on the test sets from September to December 2023

As summarized in Table 7, the misclassification errors observed in the top-performing models can be grouped into two main categories based on the direction of the prediction error:

i) False positives (non-spam classified as spam): the models occasionally misclassified messages characterized by extreme brevity, informality, or ambiguity (e.g., "*Aku gita*" [I am Gita], "*Halobuk*" [Hello ma'am], or single-word replies) as spam. This error is likely because these messages lack sufficient context for a high-confidence prediction. For architectures like CNN-LSTM, brief, legitimate messages such as *'Aku gita'* may trigger local pattern detectors (CNNs) that have learned to associate short, non-grammatical inputs with low-effort spam, failing to capture the benign sequential context that the LSTM layer would typically analyze.

ii) False negatives (spam classified as non-spam): more importantly, the models struggled to classify sophisticated spam that adeptly mimics human conversation or legitimate system notifications. Instances such as "*Kakak dm*" [Sister, DM me] employ informal, personable language to circumvent filters, whereas messages like "Welcome to Gboard clipboard..." masquerade as benign system alerts. This evidence demonstrates a clear evolution in spamming techniques toward social engineering and contextual camouflage to evade detection, which presents an ongoing challenge that necessitates models capable of discerning not just lexical content, but also underlying intent.

This analysis indicates that while deep learning models possess considerable predictive power, their accuracy can be compromised when confronted with messages that are either context-deficient or intentionally deceptive. In this respect, the DistilBERT model demonstrated a marginal advantage over CNN-LSTM by exhibiting a lower incidence of false negatives. This is a crucial distinction for practical implementation. A false negative (letting spam through) directly degrades the user experience and fails to solve the core problem. In contrast, a false positive (blocking a legitimate message) is also undesirable but can be mitigated with a user feedback mechanism. DistilBERT's bidirectional context-awareness appears to give it a slight edge in identifying these camouflaged spam messages, making it a more robust choice for ensuring a high-quality, secure user experience, which is often the paramount objective.

Table 7. Representative examples of false positive and false negative misclassifications by the top models

Misclassification type	Model	Example message	Analysis/reason for error
False positive (non-spam as spam)	CNN-LSTM/DistilBERT	"*Aku gita*", "*cs*", "*Halobuk*"	Legitimate messages are too brief, informal, or lack context, mimicking low-effort spam.
False negative (spam as non-spam)	CNN-LSTM/DistilBERT	"*Aku rio*", "*Kakak dm*", "*Selamat datang di papan klip Gboard...*"	Spam is camouflaged, mimicking friendly conversation or benign system notifications to evade detection.

3.4. Estimated impact on fallback rate

A primary objective of this research was to reduce the chatbot's fallback rate, which stood at 33% (6 million out of 18 million messages), to a target of 15% or less. The analysis in sub-section 3.2 identified the CNN-LSTM model as the top performer in terms of overall accuracy and F1-score. Therefore, we can now simulate the implementation of this specific model to estimate its direct impact on the overall fallback rate.

Based on the manual annotation of the 170,000-message sample, an estimated 29.4% of the fallback-inducing messages were identified as spam (49,952 spam messages out of 169,960 total samples). Extrapolating this to the entire population, approximately 1.764 million of the 6 million fallback messages are spam, while the remaining 4.236 million are legitimate, non-spam queries that the chatbot failed to comprehend. By implementing the CNN-LSTM model (recall: 0.91, precision: 0.92), we can project the following:

i) Spam filtered (true positives): the model would correctly identify and filter 1.605 million spam messages (1.764 million*0.91 recall). These messages would be prevented from triggering a fallback.

ii) Spam missed (false negatives): approximately 0.159 million spam messages (1.764 million*(1-0.91)) would be missed by the filter and would still result in a fallback.

iii) Legitimate fallbacks (true negatives): the model must also process the 4.236 million legitimate fallback queries. Based on the model's performance on the test sets, it has a false positive rate (FPR) of 4.52%. This means it would incorrectly filter 0.191 million legitimate messages (4.236 million*0.0452). The remaining 4.045 million legitimate queries (true negatives) would correctly pass through the filter and would still cause a fallback, as the model is designed to detect spam, not to fix the chatbot's underlying comprehension issue.

The new estimated total fallback count would therefore be the sum of missed spam (0.159 million) and the legitimate failures that passed the filter (4.045 million), resulting in 4.204 million fallbacks. This intervention would reduce the overall fallback rate from 33% (6 million/18 million) to an estimated 23.35%

(4.204 million/18 million). While this represents a significant reduction in fallback incidents, it does not meet the ambitious 15% target. This finding is critical, as it demonstrates that spam mitigation is only one part of the solution; a substantial portion of the fallback problem (4.045 million messages) is due to legitimate user queries that the chatbot fails to understand. To address the rigor of this estimation, a sensitivity analysis was performed to understand how the final fallback rate would change based on small variations in the model's performance and data assumptions. This analysis tests the stability of the 23.35% estimate by varying the two most critical metrics: the model's recall (its ability to catch spam) and its FPR (its error rate on legitimate messages). The sensitivity analysis in Table 8 demonstrates that even with a 10% fluctuation in model performance, the final fallback rate remains relatively stable (between 22.67% and 24.13%). This finding reinforces the paper's main conclusion for this section: while the spam filter is highly effective, the majority of the fallback problem (over 4 million messages) is caused by legitimate, non-spam queries that the chatbot's core intent recognition module cannot understand.

Table 8. Sensitivity analysis of estimated fallback rate

Scenario	Assumed recall	Assumed FPR	Estimated new fallback rate
Pessimistic (10% worse performance)	0.82 (from 0.91)	4.97% (from 4.52%)	24.13%
Baseline (as calculated)	0.91	4.52%	23.35%
Optimistic (10% better performance)	0.99 (from 0.91)	4.07% (from 4.52%)	22.67%

4. CONCLUSION

This research successfully engineered and empirically evaluated a suite of machine learning and deep learning models for the purpose of spam detection within WhatsApp chatbot communications. The findings from this research conclusively demonstrate that deep learning models, particularly the CNN-LSTM and DistilBERT architectures, are significantly more effective than traditional machine learning algorithms for this classification task. These models exhibited high levels of accuracy and a sophisticated capacity for understanding the contextual nuances inherent in chatbot messages. In terms of practical impact, a simulation based on the top-performing CNN-LSTM model (accuracy 0.94, recall 0.91) projects that its implementation would reduce the chatbot's overall fallback rate from 33% down to an estimated 23.35%. This directly addresses the research problem by demonstrating a significant mitigation of spam-induced fallbacks. However, this analysis also reveals that a large volume of fallbacks is caused by legitimate, non-spam queries, indicating that achieving the 15% target requires not only spam filtering but also substantial improvements to the chatbot's core intent recognition module. From an informatics and engineering perspective, the strong performance of the more computationally efficient CNN-LSTM and DistilBERT models is highly significant. This efficiency is critical for real-time processing in high-throughput smart customer service systems, as low latency is essential for user satisfaction. Furthermore, this has direct system-level integration implications relevant to electrical engineering. Unlike larger models (like BERT-base or XLM-ROBERTa) that often require heavy cloud resources, these efficient architectures are ideal candidates for edge deployment. They are viable for implementation in embedded natural language processing (NLP) modules on local servers, which would reduce network dependency, minimize application programming interface (API) latency, and lower operational costs. Building on these findings, avenues for future research include the exploration and fine-tuning of other advanced transformer-based models specifically trained on Indonesian, such as Indonesian BERT (IndoBERT) and a lite BERT (ALBERT), or the development of novel hybrid architectures designed to more effectively counter the evolving tactics of spammers. Furthermore, the expansion of the training dataset and the execution of real-time A/B testing to validate the simulated fallback reduction would constitute valuable next steps for this research.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Satrio Sadewo	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Amalia Zahra	✓	✓		✓	✓					✓		✓		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are proprietary and are not publicly available due to privacy and commercial sensitivity.

REFERENCES

[1] S. Kemp, "Digital 2023: global overview report," *datareportal.com*. Accessed: Nov. 04, 2023. [Online]. Available: https://datareportal.com/reports/digital-2023-global-overview-report

[2] C. V. Misischia, F. Poecze, and C. Strauss, "Chatbots in customer service: their relevance and impact on service quality," *Procedia Computer Science*, vol. 201, pp. 421–428, 2022, doi: 10.1016/j.procs.2022.03.055.

[3] V. Kaushal and R. Yadav, "Learning successful implementation of chatbots in businesses from B2B customer experience perspective," *Concurrency and Computation: Practice and Experience*, vol. 35, no. 1, Jan. 2023, doi: 10.1002/cpe.7450.

[4] L. Anaya, A. Braizat, and R. Al-Ani, "Implementing AI-based chatbot: benefits and challenges," *Procedia Computer Science*, vol. 239, pp. 1173–1179, 2024, doi: 10.1016/j.procs.2024.06.284.

[5] A. K. Shrivas, A. K. Dewangan, S. M. Ghosh, and D. Singh, "Development of proposed ensemble model for spam e-mail classification," *Information Technology and Control*, vol. 50, no. 3, Sep. 2021, doi: 10.5755/j01.itc.50.3.27349.

[6] K. Debnath and N. Kar, "Email spam detection using deep learning approach," in *2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON)*, May 2022, pp. 37–41, doi: 10.1109/COM-IT-CON54601.2022.9850588.

[7] S. Kaddoura, G. Chandrasekaran, D. E. Popescu, and J. H. Duraisamy, "A systematic literature review on spam content detection and classification," *PeerJ Computer Science*, vol. 8, Jan. 2022, doi: 10.7717/peerj-cs.830.

[8] T. Jain, P. Garg, N. Chalil, A. Sinha, V. K. Verma, and R. Gupta, "SMS spam classification using machine learning techniques," in *2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, 2022, pp. 273–279, doi: 10.1109/Confluence52989.2022.9734128.

[9] M. Raza, N. D. Jayasinghe, and M. M. A. Muslam, "A comprehensive review on email spam classification using machine learning algorithms," in *2021 International Conference on Information Networking (ICOIN)*, Jan. 2021, pp. 327–332, doi: 10.1109/ICOIN50884.2021.9334020.

[10] S. Gadde, A. Lakshmanarao, and S. Satyanarayana, "SMS spam detection using machine learning and deep learning techniques," in *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Mar. 2021, pp. 358–362, doi: 10.1109/ICACCS51430.2021.9441783.

[11] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North*, 2019, pp. 4171–4186, doi: 10.18653/v1/N19-1423.

[12] B. Sonare, G. J. Dharmale, A. Renapure, H. Khandelwal, and S. Narharshettiwar, "E-mail spam detection using machine learning," in *2023 4th International Conference for Emerging Technology (INCET)*, May 2023, pp. 1–5, doi: 10.1109/INCET57972.2023.10170187.

[13] V. I. D. Rosario, B. D. P. Fernandez, and D. A. Padilla, "Email spam classification using DistilBERT," in *2023 IEEE 15th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, 2023, pp. 1–6, doi: 10.1109/HNICEM60674.2023.10589211.

[14] V. S. Vinitha, D. K. Renuka, and L. A. Kumar, "Long short-term memory networks for email spam classification," in *2023 International Conference on Intelligent Systems for Communication, IoT and Security (ICISCoIS)*, 2023, pp. 176–180, doi: 10.1109/ICISCoIS56541.2023.10100445.

[15] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*, vol. 103. in Springer Texts in Statistics, vol. 103. New York, United States: Springer New York, 2013, doi: 10.1007/978-1-4614-7138-7.

[16] R. E. Schapire, "The boosting approach to machine learning: an overview," in *Nonlinear Estimation and Classification*, New York, United States: Springer, 2003, pp. 149–171, doi: 10.1007/978-0-387-21579-2_9.

[17] K. Potdar, T. S. Pardawala, and C. D. Pai, "A comparative study of categorical variable encoding techniques for neural network classifiers," *International Journal of Computer Applications*, vol. 175, no. 4, pp. 7–9, Oct. 2017, doi: 10.5120/ijca2017915495.

[18] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python*. Sebastopol, United States: O'Reilly Media, Inc., 2009.

[19] P. Wang, B. Xu, J. Xu, G. Tian, C.-L. Liu, and H. Hao, "Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification," *Neurocomputing*, vol. 174, pp. 806–814, Jan. 2016, doi: 10.1016/j.neucom.2015.09.096.

[20] M. Haroon, "Comparative analysis of stemming algorithms for web text mining," *International Journal of Modern Education and Computer Science*, vol. 10, no. 9, pp. 20–25, Sep. 2018, doi: 10.5815/ijmecs.2018.09.03.

[21] N. N. A. Sjarif, N. F. M. Azmi, S. Chuprat, H. M. Sarkan, Y. Yahya, and S. M. Sam, "SMS spam message detection using term frequency-inverse document frequency and random forest algorithm," *Procedia Computer Science*, vol. 161, pp. 509–515, 2019, doi: 10.1016/j.procs.2019.11.150.

[22] Y. Wu, M. Yuan, S. Dong, L. Lin, and Y. Liu, "Remaining useful life estimation of engineered systems using vanilla LSTM neural networks," *Neurocomputing*, vol. 275, pp. 167–179, Jan. 2018, doi: 10.1016/j.neucom.2017.05.063.

[23] J. Li, "A comparative study of LSTM variants in prediction for Tesla's stock price," *BCP Business & Management*, vol. 34, pp. 30–38, Dec. 2022, doi: 10.54691/bcpbm.v34i.2861.
[24] N. Chen, "Exploring the development and application of LSTM variants," *Applied and Computational Engineering*, vol. 53, no. 1, pp. 103–107, Mar. 2024, doi: 10.54254/2755-2721/53/20241288.
[25] S. K. Akpatsa *et al.*, "Online news sentiment classification using DistilBERT," *Journal of Quantum Computing*, vol. 4, no. 1, pp. 1–11, 2022, doi: 10.32604/jqc.2022.026658.
[26] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, Nov. 2020, doi: 10.1016/j.neucom.2020.07.061.
[27] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *International Journal of Data Mining & Knowledge Management Process*, vol. 5, no. 2, pp. 1–11, Mar. 2015, doi: 10.5121/ijdkp.2015.5201.
[28] S. Laqtib, K. E. Yassini, and M. L. Hasnaoui, "A technical review and comparative analysis of machine learning techniques for intrusion detection systems in MANET," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 3, pp. 2701–2709, Jun. 2020, doi: 10.11591/ijece.v10i3.pp2701-2709.

BIOGRAPHIES OF AUTHORS

Satrio Sadewo is currently pursuing a master of Computer Science degree at the BINUS Graduate Program, Bina Nusantara University, Jakarta, Indonesia. He received his Bachelor of Engineering (S.Kom.) in Informatics from Universitas Islam Indonesia (UII) in 2015. Alongside his studies, he currently works as a professional at an Indonesian Government Agency. His primary research interests include artificial intelligence (AI), machine learning, and deep learning. This paper is submitted as a partial fulfillment of the requirements for his master's degree graduation. He can be contacted at email: satrio.sadewo@binus.ac.id.

Amalia Zahra holds a Ph.D. in Computer Science from the School of Computer Science and Informatics, University College Dublin (UCD), Ireland, obtained in 2014. She received her bachelor's degree in Computer Science from the University of Indonesia (UI), Indonesia, in 2008. She is currently a lecturer and quality coordinator at the master of Computer Science Program, Bina Nusantara University, a position she has held since 2017. Prior to this, she served as a lecturer and researcher at the Faculty of Computer Science, University of Indonesia. Her research interests include speech processing, speech recognition, speaker recognition, spoken language identification, machine learning, deep learning, big data analytics, and computational intelligence. She can be contacted at email: amalia.zahra@binus.ac.id.