Mohammed and Ahmed
Journal of Electrical Systems and Inf Technol (2024) 11:26
https://doi.org/10.1186/s43067-024-00151-3

Journal of Electrical Systems and Information Technology

RESEARCH Open Access

A semantic-based model with a hybrid feature engineering process for accurate spam detection

Chira N. Mohammed[1*] and Ayah M. Ahmed[1]

*Correspondence:
chira.mohammed@uoz.edu.krd

[1] Department of Computer Science, University of Zakho, Zakho, Iraq

Abstract

Detecting spam emails is essential to maintaining the security and integrity of email communication. Existing research has made significant progress in developing effective spam detection models, but challenges remain in improving classification performance and adaptability to evolving spamming techniques. In this study, we propose a novel spam detection model with a comprehensive feature engineering approach that combines term frequency-inverse document frequency (TF-IDF) vectorizer and word embedding features to optimize the feature space. Our contribution lies in integrating semantic-based word embeddings, leveraging pre-existing knowledge to capture the semantic meaning of words and enhance the representation of email texts. To identify the most suitable word embedding technique for our model, we evaluated GloVe, Word2Vec, and FastText. GloVe was selected for its better performance, which is the result of its pre-training on a large and diverse text corpus. Furthermore, the model was evaluated without word embeddings, which did not exhibit the same effectiveness level as our word embedding-based model. Additionally, we utilized the support vector machine as a classifier and hyperparameter tuning technique to identify our model's most effective parameter values. The proposed model was tested on two datasets. The experimental results showed that our model outperformed the other models discussed in the literature, achieving an accuracy of 99.5% on the SpamAssassin dataset, and 99.28% on the Enron-Spam dataset.

Keywords: Spam detection, Feature engineering, TF-IDF, Word embeddings, Feature selection, SVM

Introduction

Spam email, or unsolicited bulk email, continues to be a significant challenge in the field of email communication. These unwanted messages consume valuable network resources, time, and effort and pose serious security risks, such as spreading malware and phishing attacks [1]. Therefore, an effective spam email detection model is essential for protecting users against these risks and ensuring email systems function properly.

Over the years, methods based on machine learning have become more popular. These methods leverage the power of computational algorithms to automatically learn discriminative patterns and classify emails as either spam or legitimate (non-spam) [2].

Moreover, the semantic-based method has demonstrated its effectiveness in enhancing performance across various natural language processing tasks [3]. Therefore, feature engineering which involves transforming raw email data into a suitable representation that captures relevant information by using semantics plays an essential role in the effectiveness of machine learning-driven models for detecting spam [4].

In this study, a thorough feature engineering model is proposed by combining two methods: TF-IDF, a widely used method for text representation that captures the importance of words in emails [5], and pre-trained global vectors (GloVe) word embedding for Word Representation, which enhances the representation of email texts by incorporating the semantic meaning of words [6]. The main contributions of this study are as follows:

- Novel feature engineering method combining TF-IDF and GloVe word embedding for enhanced email text representation.
- Selection of GloVe over Word2Vec and FastText for improved semantic relationship encoding in email classification.
- Incorporation of mutual information-based feature selection to optimize feature space.
- Utilization of SVM classifier for effective handling of high-dimensional feature spaces.
- Conducting hyperparameter tuning to select the best parameter values for the TF-IDF vectorizer, SVM classifier, and feature selection process, which optimizes its performance and improves its adaptability to varying datasets.
- Experimented on SpamAssassin and Enron-Spam datasets, achieving superior accuracy, surpassing existing spam detection models reported in the literature.

Continuing with this paper, we organize it as follows: Sect. "Literature review" provides a comprehensive review of related work in spam email detection. Sect. "The proposed model methodology" describes the methodology of the proposed model. Sect. "Results and discussion" includes the outcomes of our experiments and presents a comprehensive evaluation of our models in comparison with other established models from the existing literature. Finally, the paper is concluded in Sect. "Conclusion".

Literature review

Throughout the years, researchers have dedicated substantial efforts to developing effective spam detection models using diverse approaches [7]. These models leverage various techniques, including machine learning, natural language processing, statistical analysis, and network analysis, to identify and distinguish spam from legitimate messages [8]. In this literature review, we explore recent advancements in spam detection models. By synthesizing and analyzing the literature, we seek to contribute to the broader understanding of spam detection and assist researchers and practitioners in deciding which spam detection model to use and implementing it.

Ghourabi et al. [9] proposed a hybrid deep learning model (CNN-LSTM) for SMS spam detection in mixed Arabic and English messages. An accuracy of 98.37% is achieved, outperforming traditional machine learning algorithms. Their model combines CNN for identifying common spam words and LSTM for capturing long-term

dependencies. Their study contributes a labeled Arabic SMS dataset, addressing the challenge of collecting significant data for Arabic SMS spam studies. The CNN-LSTM model effectively filters spam messages and improves smartphone security. The results highlight the potential of deep learning techniques in SMS spam detection and the need for robust models in handling mixed-language environments.

Liu et al. [10] presented a study on detecting SMS spam messages using a modified transformer model based on the vanilla transformer. The findings contribute to advancing spam detection techniques using deep learning architectures consisting of positional encoding, encoder layers with self-attention, decoder layers with multi-head attention, fully-connected linear layers, and a final activation function for classification. Their model is evaluated on the SMS Spam Collection v.1 and UtkMl's Twitter datasets, and they compared its performance with various machine learning classifiers and LSTM deep learning approaches. According to their experiments, their model attains 98.92% accuracy on SMS Spam Collection v.1. While, when tested on UtkMl's Twitter dataset, their model showcases notable improvement in all evaluated aspects.

Zamir et al. [11] proposed a feature-centric spam email detection model (FSEDM) that incorporates content, sentiment, semantic, user, and spam-lexicon features. It aims to enhance classification accuracy by considering sentiment features alongside other proposed features. FSEDM model utilizes diverse supervised learning techniques and feature selection methods to prioritize significant features and to determine whether emails are spam or not. Results from their experiment demonstrated the competitive performance of the proposed model, with a deep neural network (DNN) and sentiment feature achieving a classification accuracy of up to 97.20%.

Douzi et al. [12] presented a novel hybrid method that utilizes the neural network model known as paragraph vector-distributed memory (PV-DM), for robust spam filtering. They claim that Bag-of-Words (BOW) is commonly used but it has limitations. Empirical tests on Enron and Ling spam datasets confirm their proposed method's superiority over PV-DM and BOW. By integrating global and local contexts using PV-DM and TF-IDF, the approach achieves optimal results, achieving a 98.27% overall accuracy on Ling spam datasets. Their strategy establishes an effective filter for robust email classification.

Yerima and Bashar [13] introduced a system that uses a semi-supervised innovative detection approach based on One Class SVM (OC-SVM) for detecting SMS spam. Their system served as an anomaly detector, learning from normal SMS messages without requiring labeled spam data. Evaluation on a benchmark dataset demonstrates its superiority over traditional supervised machine learning methods, achieving a 98.00% overall accuracy. Their system overcomes the challenges of imbalanced datasets by utilizing only non-spam data. The method involved preprocessing, integer encoding, and low dimensional vector embedding for OC-SVM training, resulting in excellent performance compared to bag-of-words supervised models.

Saidani et al. [14] proposed a two-level semantic analysis approach for spam detection in emails. Emails are divided into distinct domains at the first level to enable the categorization of spam that is distinctive to each domain. In the second level, each domain's semantic features are extracted, using a combination of manually specified and automatically-extracted rules. The experimental findings demonstrated that the performance

of their approach in each domain categorization task was evaluated using metrics such as precision, recall, accuracy, and F1-measure, and it consistently showed high performance in comparison with various state-of-the-art methods that rely on bag-of-words (BoW) and latent semantic analysis.

Siddique et al. [15] introduced an automated approach for detecting Urdu spam emails. Their study utilizes various machine learning and deep learning algorithms such as SVM, Naive Bayes, CNN, and LSTM, for email content detection and categorization. The LSTM model demonstrated the highest accuracy of 98.40%, outperforming other models. Their study emphasizes the importance of automated approaches for detecting Urdu spam emails and highlights the efficiency and accuracy of deep learning models, particularly LSTM.

Fatima et al. [16] presented a machine learning-based approach for classifying spam emails. Their study utilizes two feature extraction modules, Count-Vectorizer and TF-IDF-Vectorizer, and evaluates various ML algorithms such as Naive Bayes, logistic regression, extra tree, SGD, XG-boost, SVM, RF, and MLP. Hyperparameter tuning is applied to optimize the classifiers. Their model achieves high accuracy on different datasets, outperforming other state-of-the-art models. The research emphasizes the importance of preprocessing, feature selection, and hyperparameter tuning in improving classification results.

In our study, a distinguishing characteristic lies in the feature engineering approach, which combines techniques such as TF-IDF and word embedding features, as well as incorporating an optimal feature selection process to optimize the feature space and utilizing a powerful classifier to develop a robust and accurate spam detection model. By incorporating pre-trained GloVe word embeddings as a semantic-based technique, the code effectively captures the semantic meaning of words, enhancing the representation of email texts. Furthermore, the model utilizes hyperparameter tuning to select the best parameter values for the TF-IDF vectorizer, SVM classifier, and feature selection process. In comparison with the most recent spam detection models documented in the literature, the proposed model exhibited a better accuracy performance. These feature engineering techniques collectively contribute to improved classification performance, generalization, stability, and adaptability, making the code a powerful tool for spam email detection tasks.

The proposed model methodology

This section presents the methodology followed by the proposed model which is illustrated in Fig. 1. The model development process began with the acquisition of diverse datasets, including the SpamAssassin Dataset and the Enron-Spam dataset. These datasets underwent preprocessing steps such as lowercase conversion, punctuation removal, stop word elimination, and lemmatization for text normalization. Following preprocessing, the data were split into training and testing sets. Feature engineering was then conducted, involving the extraction of features such as word embeddings and TF-IDF scores, which were fused to create a comprehensive feature set. Subsequently, feature selection was performed to enhance model performance. The selected features were used to train SVM classifier. Hyperparameter tuning was then employed to optimize the model's performance. Finally, evaluation metrics, including accuracy, precision, recall, and F1-Score,

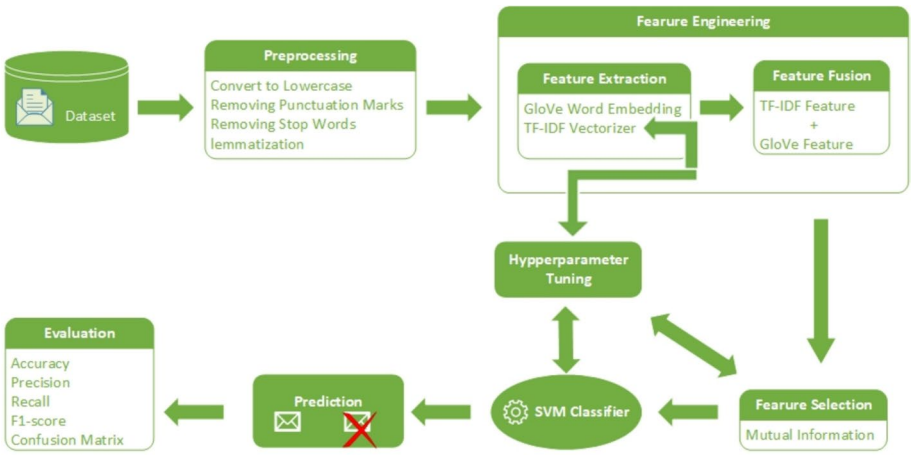

Fig. 1 Architecture of the proposed spam detection model

along with confusion matrix analysis, were utilized to assess the model's effectiveness in spam email detection, ensuring robust email communication security. Each individual step represented in Fig. 1 is described in detail in the following subsections.

Dataset description

In this study, two distinctive datasets have been used, each contributing valuable insights and a unique perspective to the evaluation of our model. Both datasets were obtained from established repositories, ensuring the authenticity and relevance of the data. The first dataset is a collection of emails taken from Apache SpamAssassin's public datasets, which is available for public access on the Kaggle platform named Spam or Not Spam Dataset [17]. This dataset is characterized by its unique distribution of spam and ham emails, it contains 2,500 ham and 500 spam emails. The dataset provides a valuable resource for training and evaluating spam classification models. The second dataset, the Enron-Spam dataset [18], includes an extensive collection of 33,716 emails; this dataset is a comprehensive resource for evaluating the performance of our model. One of the standout features of the Enron-Spam dataset is the well-balanced distribution of spam and ham emails. It comprises 17,171 spam emails and 16,545 ham emails, creating an environment that mirrors real-world email communication more closely. The utilization of both datasets allowed us to conduct a comprehensive assessment of our model, considering different numbers of emails and various email distribution scenarios.

Data preprocessing

To prepare the email messages for analysis, a series of preprocessing steps was performed. First, the text was converted to lowercase. Punctuation marks were removed using regular expressions, and stop words were eliminated. Additionally, lemmatization was applied to reduce words to their base form and to improve text normalization.

Data splitting

The data were divided into two main subsets: 80% training set and 20% testing set. The model was trained using the training set, allowing it to learn from the data and adjust its

parameters accordingly. The testing set was kept separate from the training phase and solely used for evaluating the model's performance. This division ensured that the model's effectiveness was rigorously assessed on unseen data, providing a reliable indication of its ability to accurately detect spam emails.

Feature engineering

Feature engineering is a critical phase in our spam detection model, consisting of two essential steps: feature extraction and feature combination. These steps collectively enable the transformation of raw text data into numerical representations, empowering our model to make accurate predictions.

Feature extraction

Our model contains two feature extraction processes: word embeddings and TF-IDF features. Pre-trained GloVe word embedding model was utilized to capture the semantic meaning and relationships between words by placing them in a high-dimensional vector space, specifically the "glove-wiki-gigaword-300" model, which provides 300-dimensional word vectors to convert each word or token in the text to a dense vector representation [6]. This process includes tokenization, extraction of word embeddings for each word, and averaging these embeddings to obtain a single vector representing the entire email text.

We employ TF-IDF features to quantify word importance in email texts. TF-IDF scores are calculated for each word in the email texts, generating numerical feature vectors that reflect the importance of words within individual emails and across the entire dataset. After these processes, the obtained TF-IDF feature vectors and word embeddings are correctly converted into arrays, making them suitable for further processing and integration into the feature set.

Feature fusion

To create a comprehensive feature set for training our spam detection model, we employ feature combinations. This process involves the fusion of the two arrays of TF-IDF feature vectors and word embeddings, effectively merging the semantic context captured by embeddings with the statistical characteristics of TF-IDF features. Through this fusion and the utilization of arrays, our model optimizes data processing, making it a valuable tool for email communication security, and gains a multifaceted view of email content, resulting in improved detection accuracy and effectiveness.

Feature selection

To reduce dimensionality and improve model performance, feature selection was performed using mutual information [19]. Mutual information is a statistical metric that measures the dependence between two variables [20]. The feature selection process is executed on the combined feature set through the utilization of the SelectKBest algorithm, combined with the mutual information classif score function. This combination facilitates the selection of the top-K features that exhibit the highest mutual information with the target variable, which in our context represents the classification of emails into spam or ham.

Model training

SVM classifier was trained on the selected features from the training set. It is a popular supervised learning algorithm known for its effectiveness in binary classification tasks. The trained SVM model was then used to predict the spam/ham labels for the test set. The objective of utilizing the SVM classifier is to achieve high classification accuracy and generalization performance.

Hyperparameter tuning

Hyperparameter tuning significantly enhanced the performance of our spam email detection model, achieving remarkable accuracy and adaptability across diverse datasets by systematically exploring the hyperparameter space, utilizing the pipeline module, which streamlines the process by combining multiple steps into a single workflow [21], and integrating it with the GridSearchCV module, which systematically explores a range of hyperparameter combinations [22], along with cross-validation techniques that assess model performance on various data subsets [23], allowed us to efficiently explore and optimize key parameters within defined ranges while keeping other parameters at their default values.

The TF-IDF vectorizer's parameters underwent systematic exploration to discern their effects on feature extraction and representation. This included max_features, signifying the maximum count of features; ngram_range, defining the range of n-grams; sublinear_tf, indicating potential sublinear term frequency scaling; use_idf, determining inverse document frequency IDF incorporation in feature weighting; smooth_idf, governing IDF weight smoothing; and min_df, specifying the minimum document frequency for term consideration.

- max_features: Ranging from 5000 to 10,000. (Chosen: 5000).
- ngram_range: Explored with unigrams and bigrams: [(1, 1), (1, 2)]. (Chosen: (1, 1)).
- sublinear_tf: True, False. (Chosen: True).
- use_idf: True, False. (Chosen: True).
- smooth_idf: True, False. (Chosen: True).
- min_df: Ranging from 2 to 5. (Chosen: 2).

Considering the SelectKBest feature selection method, parameters underwent systematic exploration to evaluate their impact on feature selection. This involved the assessment of score_func, determining the scoring function used for feature selection, and k, representing the number of top features to select.

- score_func: Explored mutual information mutual_info_classif, chi-squared chi2, and Analysis of Variance F-value f_classif. (Chosen: mutual_info_classif).
- k: Varied within the range of 200–400. (Chosen: 200).

For the SVM classifier, parameters were meticulously explored to optimize its performance. This included C, representing the regularization parameter that regulates the penalty for misclassification; kernel, determining the type of kernel function used; and gamma, which influences the kernel coefficient.

- C: Explored values ranged from 0.1 to 10. (Chosen: 10).
- kernel: Options explored included linear, radial basis function (RBF), and sigmoid. (Chosen: RBF).
- gamma: Explored values ranged from 0.1 to 10. (Chosen: 0.1).

Experimental setup

For conducting our experiments, we leveraged scikit-learn, a popular Python machine learning library, for various tasks including feature extraction, selection, and classification. Text preprocessing was carried out using the WordNetLemmatizer and NLTK libraries. For visualization of results, we utilized matplotlib and seaborn libraries. The dataset was split into training and testing sets using scikit-learn's train_test_split function. Additionally, we incorporated TF-IDF vectorization using the TFIDF vectorizer from scikit-learn to convert text data into TF-IDF feature vectors. Hyperparameter tuning was performed using the GridSearchCV and Pipeline modules from scikit-learn. We also integrated pre-trained word embeddings from the gensim library to capture semantic meaning. Experiments were executed on a computational environment running a Windows 11 operating system, equipped with an Intel Core i5 processor and 12 GB of RAM.

Evaluation metrics

To assess the performance of the proposed model, the experiments employ the most common evaluation metrics, including accuracy, precision, recall, F1-Score, and confusion matrix. The confusion matrix is a crucial evaluation tool in the context of spam detection. It provides a comprehensive assessment of the performance of a spam detection model by displaying the predicted outcomes against the actual class labels of the data [24]. The confusion matrix is structured into four distinct quadrants: The quadrant of true positives (TP) signifies the accurate identification of spam emails. Complementary to this, the quadrant of false positives (FP) denotes non-spam emails that were inaccurately classified as spam. True negatives (TN) indicate the precise classification of non-spam emails. Conversely, the quadrant of false negatives (FN) represents the misclassification of spam emails as non-spam.

The other metrics are derived from the confusion matrix, with their mathematical expressions presented as Eqs. (1–4), respectively.

$$\text{Accuracy} = \frac{\text{Tp} + \text{TN}}{\text{TP} + \text{FP} + \text{FP} + \text{FN}} \tag{1}$$

$$\text{Precision} = \frac{\text{Tp}}{\text{TP} + \text{FP}} \tag{2}$$

$$\text{Recall} = \frac{\text{Tp}}{\text{TP} + \text{Fn}} \tag{3}$$

$$F1 - \text{Score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{4}$$

Accuracy represents the proportion of correctly classified instances [25], while precision measures how well the model can detect positive cases correctly among all the cases that it predicted as positive [26], and recall is a measure of how well a model can identify positive cases out of all those that are actually positive [27]. The F1-score indicates a balanced assessment of performance because it takes into account both recall and accuracy [1]. These evaluation metrics provide an extensive overview of the efficiency of our model in classification tasks.

Results and discussion

In this section, we conducted a thorough evaluation of the proposed model, utilizing widely accepted performance metrics. In the upcoming subsections, we will showcase the results obtained from testing the word embedding models. Subsequently, we will present and discuss the outcomes derived from the use of both datasets for our model. Finally, we will provide a comparative analysis between our proposed models with the most current existing models.

Performance based on word embedding

We conducted a comprehensive evaluation of three popular word embedding models, namely GloVe, Word2Vec, and FastText. The primary objective of the comprehensive evaluation was to select the most suitable word embedding model to be integrated into our proposed model. The GloVe model is trained on substantial Wikipedia and Gigaword data and produces 300-dimensional word vectors [6]. The FastText model originated from Wikipedia data; this model constructs 300-dimensional word vectors considering subword information [28]. The Word2Vec model originated from Google News data and also generates 300-dimensional word vectors [29]. The findings revealed that GloVe outperformed the other word embeddings, demonstrating better accuracy on both datasets, due to its unique training approach. GloVe leverages global co-occurrence statistics across the entire corpus during training [30]. This results in embeddings that effectively encode the semantic relationships between words based on their contextual usage. In contrast, Word2Vec and FastText primarily focus on local context, capturing word relationships based on neighboring words within a limited window [31]. Consequently, GloVe has been selected.

To validate the significance of word embeddings, we conducted experiments utilizing only TF-IDF without word embeddings on both datasets, which demonstrated inferior performance compared to our word embedding-based model. The accuracy performances attained through the utilization of GloVe, Word2Vec, FastText, and the absence of word embeddings, based on the SpamAssassin dataset, were recorded at 99.50%, 99.16%, 98.83%, and 97.16%, respectively. Additionally, for the Enron-Spam dataset, the accuracy rates of GloVe, Word2Vec, FastText, and the absence of word embeddings were recorded at 99.28%, 99.16%, 99.02%, and 98.01%, respectively. The outcomes of these experiments, along with additional evaluation metrics, are elaborated in Tables 1 and 2 and visualized in Figs. 2 and 3.

Table 1 Performance analysis of the proposed model with and without word embeddings models using SpamAssassin dataset

Word embeddings models	Accuracy	Precision	Recall	F1-Score
GloVe	99.50	100	96.84	98.39
Word2Vec	99.16	98.91	95.78	97.32
FastText	98.83	98.88	93.68	96.21
No embeddings	97.16	93.33	88.42	90.81

Table 2 Performance analysis of the proposed model with and without word embeddings models using Enron-Spam dataset

Word embeddings models	Accuracy	Precision	Recall	F1-Score
GloVe	99.28	98.92	99.67	99.30
Word2Vec	99.16	98.98	99.38	99.18
FastText	99.02	98.69	99.38	99.03
No Embeddings	98.01	97.15	98.97	98.05

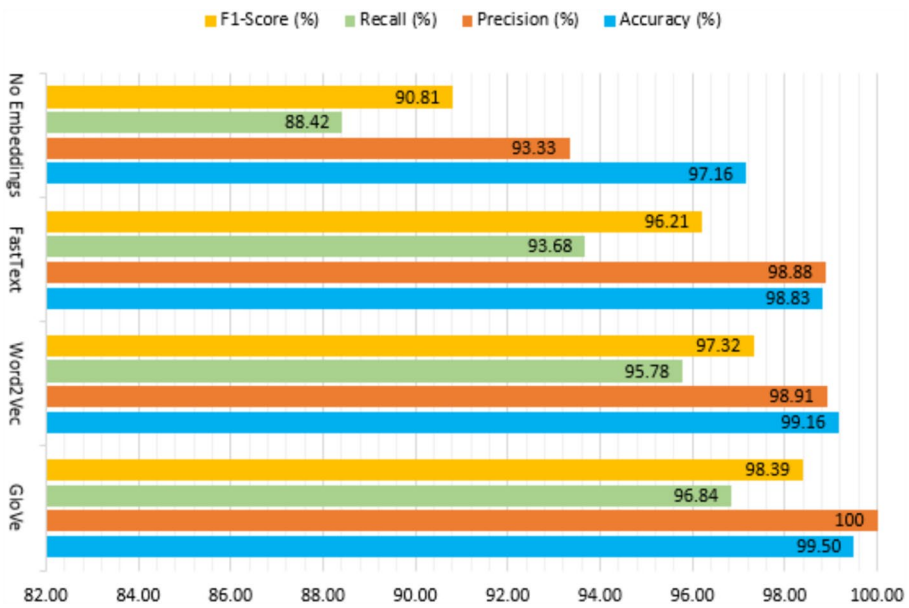

Fig. 2 Comparison of the proposed model with and without word embeddings models using SpamAssassin dataset

Overall performance of the proposed model

The developed spam detection model demonstrated exceptional performance across different datasets and evaluation metrics. When using the SpamAssassin dataset, the model achieved an impressive overall accuracy of 99.50%, accurately classifying emails as either spam or non-spam. In contrast, with the Enron-Spam datasets, the model exhibited a remarkable accuracy of 99.28%, further highlighting its robustness in email classification.

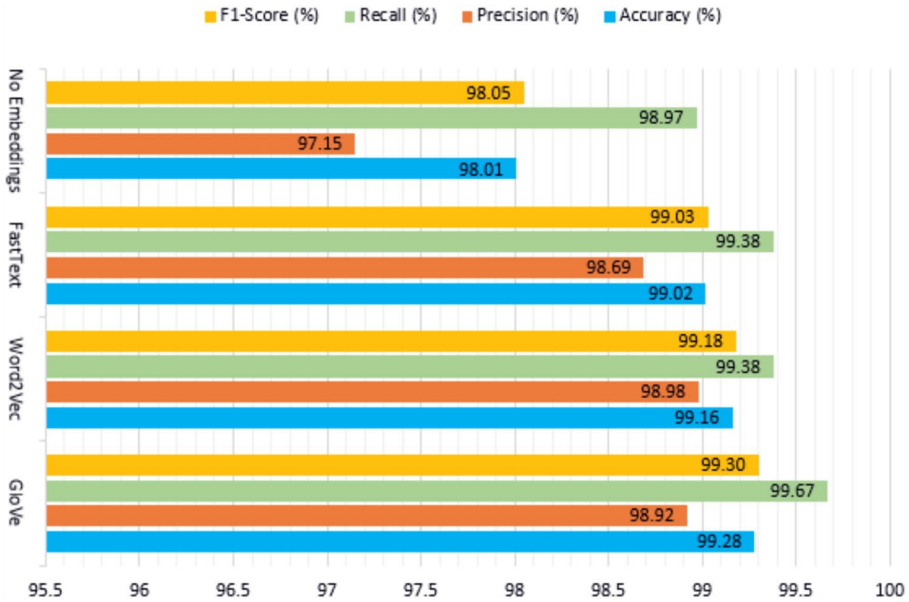

Fig. 3 Comparison of the proposed model with and without word embeddings models using Enron-Spam dataset

The precision of 100% on the SpamAssassin dataset signifies that a high proportion of the identified spam emails was indeed spam, effectively minimizing false positives. Similarly, on the Enron-Spam datasets, the precision reached 98.92%, demonstrating the model's ability to minimize the misclassification of non-spam emails as spam.

Additionally, the model achieved a remarkable recall of 96.84% when using the SpamAssassin dataset, indicating its ability to effectively capture a significant proportion of actual spam emails. On the Enron-Spam datasets, the recall rate was even higher at 99.67%, further emphasizing the model's proficiency in identifying spam emails while reducing false negatives.

Consequently, the F1-Score, calculated at 98.39% using the SpamAssassin dataset and 99.30% on the Enron-Spam datasets, showcases the balanced trade-off between precision and recall, ensuring the model's overall reliability and efficiency in identifying spam emails across different datasets. The performance results for both datasets can be found in Fig. 4, illustrating the model's consistency and effectiveness in various email classification scenarios.

The results of the confusion matrices obtained from the proposed model are illustrated in Figs. 5 and 6. As can be seen, the results obtained from the SpamAssassin dataset reveal that there were 92 instances of True Positives (TP), while there were no occurrences of False Positives (FP). True Negatives (TN) amounted to 505, and there were three instances of False Negatives (FN). In contrast, when employing the Enron-Spam dataset, the results differed, with 3408 instances of True Positives (TP), 37 occurrences of False Positives (FP), 3288 instances of True Negatives (TN), and 11 instances of False Negatives (FN). This comprehensive matrix analysis facilitates a granular evaluation of the model's performance across different classification outcomes.

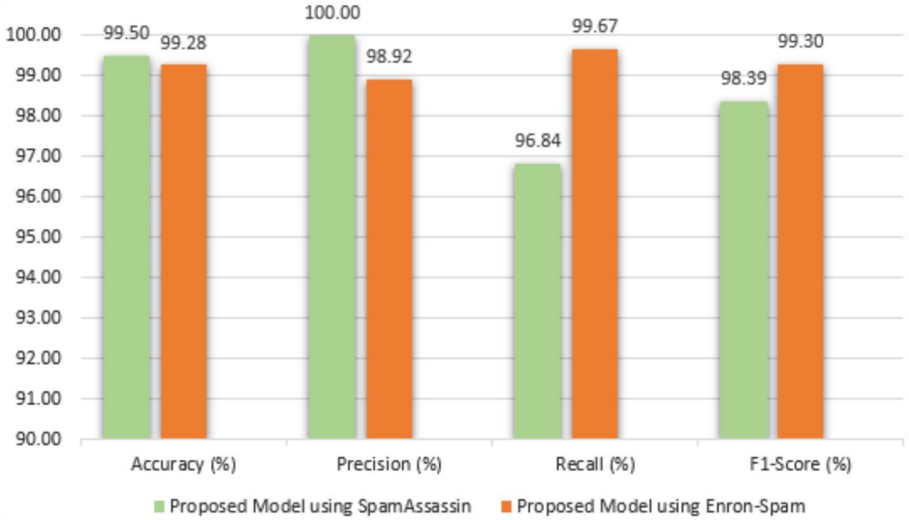

Fig. 4 Overall performance of the proposed model

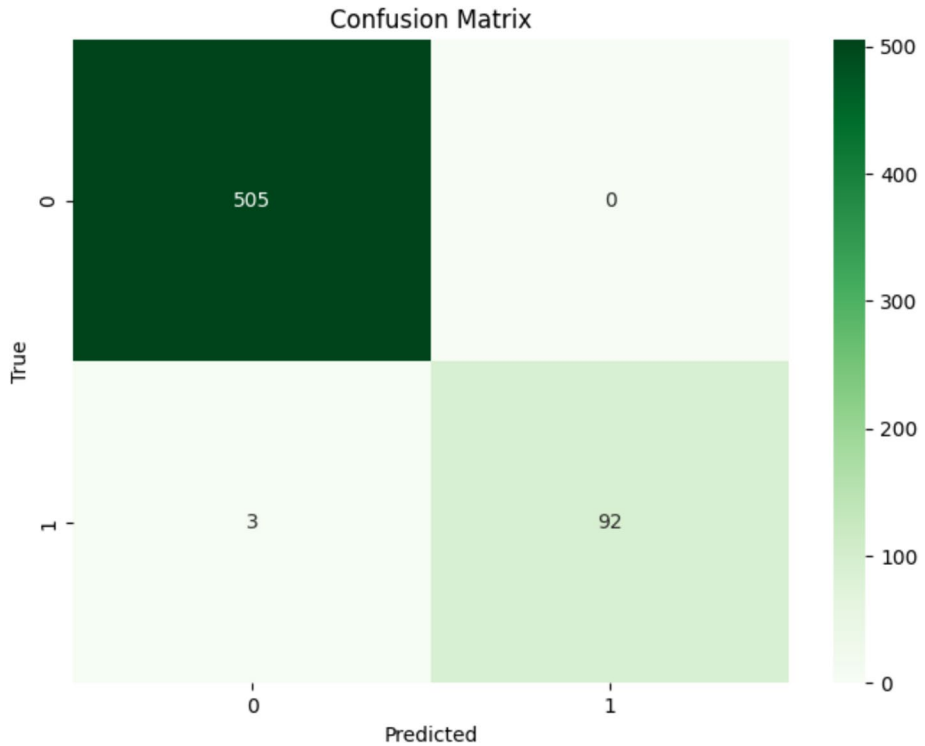

Fig. 5 Confusion matrix of the proposed model based on SpamAssassin dataset

Comparative analysis with state-of-the-art approaches

This section involves comparing the proposed model with recent models published in the literature for spam detection, aiming to assess its alignment and compatibility with existing approaches. The results obtained from the literature models represent their optimal performance achieved through the utilization of diverse approaches and

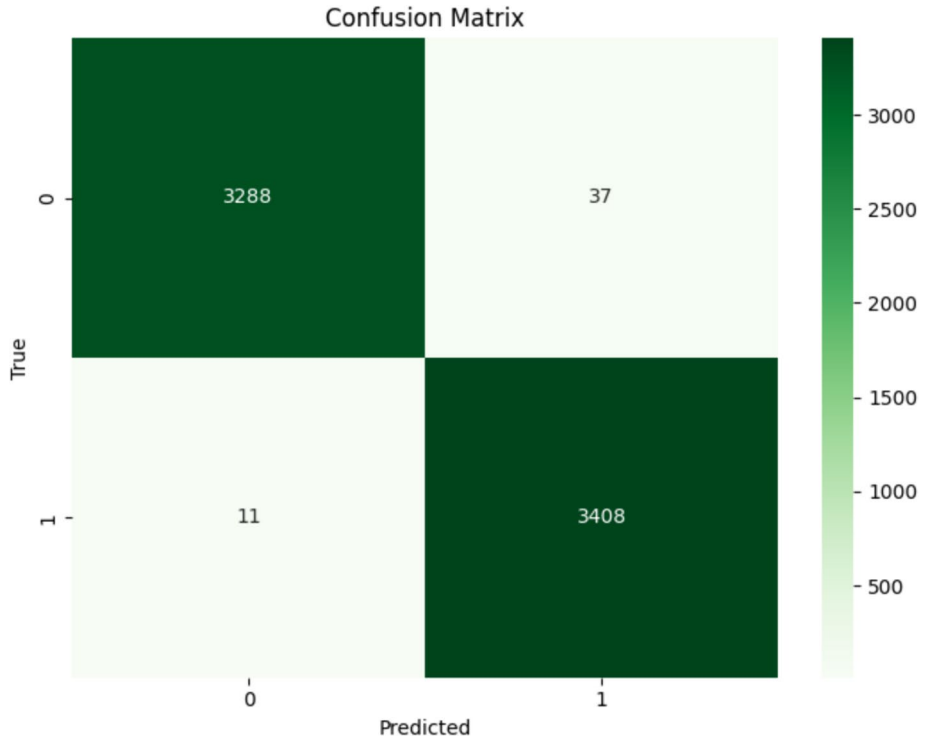

Fig. 6 Confusion matrix of the proposed model based on Enron-Spam dataset

Table 3 Performance analysis of the proposed model with relevant literature models

References	Method	Dataset type	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
[9]	CNN-LSTM	SMS Spam Collection	98.37	95.39	87.87	91.48
[10]	Vanilla transformer	SMS spam collection v.1	98.92	97.81	94.51	96.13
[11]	Sentiment feature with a deep neural network (DNN)	CSDMC2010_SPAM	97.20	94.80	95.70	95.00
[12]	Neural Network model PV-DM with TF-IDF	Ling spam	98.27	97.97	100	98.97
[13]	Semi-supervised novelty detection (OC-SVM)	Benchmark	98.00	96.80	100	98.00
proposed model	Semantic-based feature engineering model	Collection from Apache SpamAssassin	99.50	100	96.84	98.39
Proposed model	Semantic-based feature engineering model	Enron-Spam	99.28	98.92	99.67	99.30

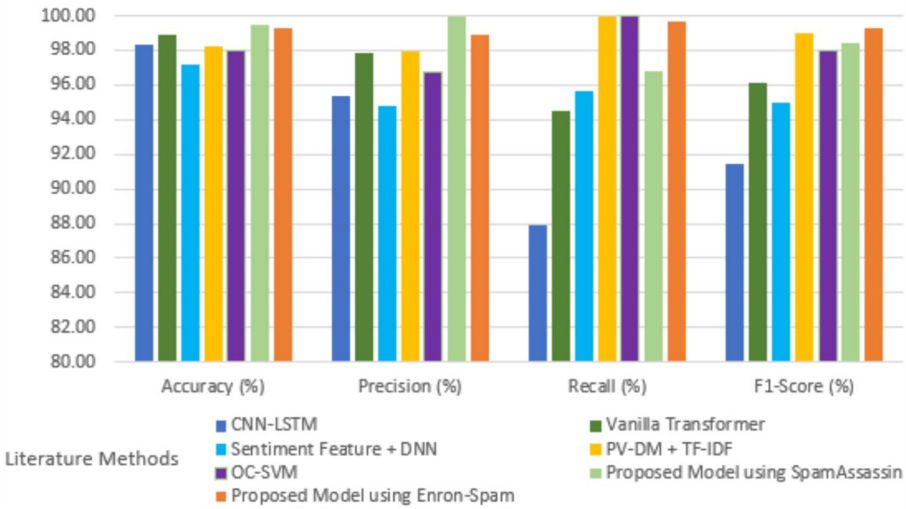

Fig. 7 Comparison of the proposed model with relevant literature models

datasets. The performance comparison was conducted based on metrics such as accuracy, precision, recall, and F1-Score. As demonstrated in Table 3 and Fig. 7, the proposed model exhibited superior performance on both datasets, surpassing all the compared literature models.

Conclusion

In this research, a semantic-based model has been presented for detecting spam emails. We compared three popular word embedding models, GloVe, Word2Vec, and FastText, to identify the most effective one for enhancing the proposed model's accuracy and efficiency. Ultimately, we selected GloVe due to its better performance in representing semantic meaning and contextual relationships in word embeddings. Moreover, the proposed model utilizes a comprehensive feature engineering approach that combines TF-IDF feature vectors and pre-trained GloVe word embeddings to effectively represent the text data. Furthermore, feature selection using mutual information was employed to select the most informative features and reduce dimensionality. Additionally, the SVM classifier has successfully trained on the selected features and evaluated its performance on the test set. To ensure that the model's parameters were fine-tuned, hyperparameter tuning techniques have been used for the TF-IDF vectorizer, SVM classifier, and feature selection process. The model demonstrated exceptional performance on the SpamAssassin dataset, achieving an impressive 99.50% accuracy, precision of 100%, recall of 96.84%, and an impressive F1-score of 98.39%. Similarly, it achieved remarkable performance on the Enron-Spam dataset, with an accuracy of 99.28%, precision of 98.92%, recall of 99.67%, and an outstanding F1-Score of 99.30%. Finally, a comparative analysis was conducted to assess the performance of the proposed model against recently published models, and the results reveal that the proposed model exhibits superior performance compared to the other models. This emphasizes the promising potential of the proposed feature engineering approach in enhancing the performance of spam email detection models.

Despite the satisfactory performance of the proposed model, it is essential to acknowledge its limitations. One notable constraint is that our comparison of word embedding models was confined to GloVe, Word2Vec, and FastText; other models may offer additional insights. Additionally, considering metrics for evaluating time and complexity could further enhance the robustness and applicability of our model.

In the future, we aim to employ bidirectional encoder representations from transformers (BERTs), which capture bidirectional context to understand the semantics of words within a sentence. We also intend to utilize a universal sentence Eecoder (USE), which can encode semantic meaning into fixed-length vectors, promising enhanced language understanding across diverse applications. Moreover, we aim to extend our comparison to include additional word embedding models.

Acknowledgements
Not applicable.

Author contributions
CNM was contributed conceptualization, software, investigation, validation, and writing—original draft. AMA was involved in conceptualization, methodology, and writing—review and editing. All authors reviewed and approved the final manuscript.

Funding
Not applicable.

Availability of data and materials
The data that support the findings of this study are available on request from the corresponding author.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
I hereby consent to the publication of my paper in Journal of Electrical Systems and Information Technology.

Competing interests
The authors declare that they have no competing interests.

Received: 25 January 2024 Accepted: 3 July 2024
Published online: 15 July 2024

References

1. Nandhini S, KS JM (2020) "Performance evaluation of machine learning algorithms for email spam detection," In 2020 international conference on emerging trends in information technology and engineering (ic-ETITE), IEEE, pp 1–4
2. Ahmed N, Amin R, Aldabbas H, Koundal D, Alouffi B, Shah T (2022) Machine learning techniques for spam detection in email and IoT platforms: analysis and research challenges. Secur Commun Netw 2022:1–19
3. Mewada A, Dewang RK (2023) A comprehensive survey of various methods in opinion spam detection. Multimed Tools Appl 82(9):13199–13239
4. Makkar A, Garg S, Kumar N, Hossain MS, Ghoneim A, Alrashoud M (2020) An efficient spam detection technique for IoT devices using machine learning. IEEE Trans Ind Inf 17(2):903–912
5. Hossain SMM, Kamal KMA, Sen A, Sarker IH, "TF-IDF feature-based spam filtering of mobile SMS using a machine learning approach," In applied intelligence for industry 4.0: Chapman and Hall/CRC, 2023, pp 162–175
6. Ghanem R, Erbay H (2023) Spam detection on social networks using deep contextualized word representation. Multimed Tools Appl 82(3):3697–3712
7. Rajesh A, Hiwarkar T (2023) Sentiment analysis from textual data using multiple channels deep learning models. J Electr Syst Inf Technol 10:56. https://doi.org/10.1186/s43067-023-00125-x
8. Aliza HY, Nagary KA, Ahmed E, Puspita KM, Rimi KA, Khater A, Faisal F (2022) "A comparative analysis of SMS spam detection employing machine learning methods," In 2022 6th international conference on computing methodologies and communication (ICCMC), IEEE, pp 916–922
9. Ghourabi A, Mahmood MA, Alzubi QM (2020) A hybrid CNN-LSTM model for SMS spam detection in arabic and english messages. Future Internet 12(9):156
10. Liu X, Lu H, Nayak A (2021) A spam transformer model for SMS spam detection. IEEE Access 9:80253–80263
11. Zamir A, Khan HU, Mehmood W, Iqbal T, Akram AU (2020) A feature-centric spam email detection model using diverse supervised machine learning algorithms. Electron Libr 38(3):633–657

12. Douzi S, AlShahwan FA, Lemoudden M, El Ouahidi B (2020) Hybrid email spam detection model using artificial intelligence. Int J Mach Learn Comput 10(2):2
13. Yerima SY, Bashar A (2022) "Semi-supervised novelty detection with one class SVM for SMS spam detection," In: 2022 29th international conference on systems, signals and image processing (IWSSIP), IEEE, pp 1–4
14. Saidani N, Adi K, Allili MS (2020) A semantic-based classification approach for an enhanced spam detection. Comput Secur 94:101716
15. Siddique ZB, Khan MA, Din IU, Almogren A, Mohiuddin I, Nazir S (2021) Machine learning-based detection of spam emails. Sci Progr 2021:1–11
16. Fatima R, Sadiq M, Ullah S, Ahmed G, Mahmood S (2023) An optimized approach for detection and classification of spam email's using ensemble methods
17. "Spam or Not Spam Dataset: a collection of emails taken from Apache SpamAssassin's public datasets." Accessed 2023. https://www.kaggle.com/datasets/ozlerhakan/spam-or-not-spam-dataset
18. "Enron-Spam datasets. Accessed 2023. " https://www2.aueb.gr/users/ion/data/enron-spam/
19. Sultana A, Islam R ((2023)) Machine learning framework with feature selection approaches for thyroid disease classification and associated risk factors identification. J Electr Syst Inf Technol 10:32. https://doi.org/10.1186/s43067-023-00101-5
20. Kaur A, Guleria K, Trivedi NK (2021) "Feature selection in machine learning: methods and comparison," In: 2021 international conference on advance computing and innovative technologies in engineering (ICACITE), IEEE, pp 789–795
21. Filippou K, Aifantis G, Papakostas GA, Tsekouras GE (2023) Structure learning and hyperparameter optimization using an automated machine learning (AutoML) pipeline. Information 14(4):232
22. Omotehinwa TO, Oyewola DO (2023) Hyperparameter optimization of ensemble models for spam email detection. Appl Sci 13(3):1971
23. Wazirali R (2020) An improved intrusion detection system based on KNN hyperparameter tuning and cross-validation. Arab J Sci Eng 45(12):10859–10873
24. Hossain F, Uddin MN, Halder RK (2021) "Analysis of optimized machine learning and deep learning techniques for spam detection," In: 2021 IEEE international IOT, electronics and mechatronics conference (IEMTRONICS), IEEE, pp 1–7
25. Madhavan MV, Pande S, Umekar P, Mahore T, Kalyankar D (2021) "Comparative analysis of detection of email spam with the aid of machine learning approaches," In: IOP conference series: materials science and engineering, 1022(1): IOP Publishing, 012113
26. Elhussein M, Brahimi S (2021) Clustering as feature selection method in spam classification: uncovering sick-leave sellers, Appl Comput Inform, 2021
27. 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), 1: IEEE, pp 358–362
28. Khasanah IN (2021) Sentiment classification using fasttext embedding and deep learning model. Procedia Comput Sci 189:343–350
29. Grohe M, word2vec, node2vec, graph2vec, x2vec: towards a theory of vector embeddings of structured data, In: proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI symposium on principles of database systems, 2020, pp 1–16
30. Asudani DS, Nagwani NK, Singh P (2022) Exploring the effectiveness of word embedding based deep learning model for improving email classification. Data Technol Appl 56(4):483–505
31. Somesha M, Pais AR (2022) Classification of phishing email using word embedding and machine learning techniques. J Cyber Secur Mobil 11:279–320

Publisher's Note