Alexandria Engineering Journal 111 (2025) 540–554

Contents lists available at ScienceDirect

Alexandria Engineering Journal

journal homepage: www.elsevier.com/locate/aej

Original article

Topic-aware neural attention network for malicious social media spam detection

Maged Nasser [a], Faisal Saeed [b,*], Aminu Da'u [c], Abdulaziz Alblwi [d], Mohammed Al-Sarem [e]

[a] Computer & Information Sciences Department, Universiti Teknologi PETRONAS, Seri Iskandar 32610, Perak, Malaysia
[b] College of Computing and Digital Technology, Birmingham City University, Birmingham B4 7XG, UK
[c] Department of Computer Science, Hassan Usman Katsina Polytechnic, Katsina State, Nigeria
[d] Department of Computer Science, Applied College, Taibah University, Saudi Arabia
[e] College of Computer Science and Engineering, Taibah University, Medina, Saudi Arabia

ARTICLE INFO

Keywords:
Spam detection
Topic modelling
Attention neural network
Malicious detection
Bidirectional encoder representations from transformers (BERT)
Online social network

ABSTRACT

Social media platforms, such as Facebook and X (formally known as Twitter), have become indispensable tools in today's society because they facilitate social discussion and information sharing. This feature makes social networks more attractive for spammers who intentionally spread fake messages, post malicious links and spread rumours. Recently, several machine learning methods have been introduced for social network malicious spam classification. However, most existing methods generally rely on handcrafted features and traditional embedding models, which are relatively less effective. Therefore, inspired by the success of the neural attention network, we propose an interactive neural attention-based method for malicious spam detection by integrating long short-term memory (LSTM), topic modelling, and the BERT technique. In the proposed approach, first, we employed the LSTM encoder, which was integrated with the Twitter latent Dirichlet allocation (LDA) model via an interactive attention mechanism to jointly learn local content and global topic representations. Second, to further learn the contextualized features of texts, the model was further integrated with the BERT technique. Last, the Softmax function was then applied at the output layer for the final spam classification. A series of experiments were conducted utilizing two real-world datasets to evaluate the model. Using dataset 1, the proposed model outperformed the baseline techniques, with average improvements in recall, precision, and F1 and accuracies of 17.54 %, 6.19 %, 11.91 %, and 12.27 %, respectively. In addition, the proposed model performed well for the second dataset and obtained average gains of 11.81 %, 4.38 %, 8.12, and 7.42 in terms of recall, precision, F1, and accuracy, respectively.

1. Introduction

Online social network (OSN) platforms such as Facebook and Twitter have become very useful for convivial microblogging [1–3]. On these platforms, people can spend most of their time reading news, posting messages, and making friends with other people [4]. The popularity of OSNs makes them more attractive for spammers who intentionally spread spam by distributing fake messages to innocent users, posting malicious links, and spreading rumours [5]. Spam contains mainly malicious Universal Resource Locators (URLs) with a financial claim and intricate content [6–8]. Online social network spammers usually utilize trending hashtags and catchphrases to attract the attention of users. Spam is a type of information that spammers actively send with the intention of misleading, disseminating false information, and generating revenue [5]. Spam that previously had little or no impact can now use OSNs to cause a massive distributed impact [1–3]. OSNs disclose all basic user information and offer follow-up functions, which enables spammers to send spam to potential target users easily and accurately and encourage dissemination [4]. Problems such as resource utilization, prolonged communication times, and bandwidth waste are associated with spam [5]. The growth rate of spam outpaced that of regular evaluations on the majority of OSNs [6]. Additionally, malware, pornography, and malicious related content are present in more than 15 % of spam. However, despite many related studies, social networks continue to have a significant amount of spam. Thus, an effective detection method for malicious spam is needed to ensure the security of OSNs for

* Corresponding author.
E-mail addresses: maged.nasser@utp.edu.my (M. Nasser), faisal.saeed@bcu.ac.uk (F. Saeed), ablwi@taibahu.edu.sa (A. Alblwi), msarem@taibahu.edu.sa (M. Al-Sarem).

https://doi.org/10.1016/j.aej.2024.10.073
Received 5 December 2023; Received in revised form 4 October 2024; Accepted 16 October 2024
Available online 29 October 2024

users. Traditional spam control methods work by determining whether a user is a spammer and then blocking him. However, spammers can open new accounts and continue spreading spam activities. Hence, it is important to detect and prevent OSNs from spam at the tweet level.

With the advancement of artificial intelligence, several approaches based on machine learning have been introduced to detect spamming activities on online social networks (OSNs). Traditional machine learning (ML) methods, such as support vector machines (SVMs), random forests (RFs), and K-nearest neighbours (KNNs), have achieved remarkable results in spam detection [4]. These methods are particularly useful for classifying spam textual content based on its characteristic attributes [1], such as the number of URLs and keywords [9]. However, despite the good results of these traditional ML methods, they occasionally suffer from computational complexity and domain dependence, which make them less reliable [4]. Thus, to address the existing shortcomings, several deep learning methods have been introduced [4, 10–12]. Most existing deep learning-based methods typically rely solely on word embeddings such as Glove [13] and Wor2vec [14] as the main semantic features for spam classification. However, these traditional word embeddings alone cannot efficiently address the task of short text classification due to their intrinsic high dimensionality and sparsity [4, 14]. To improve performance in text classification, the BERT technique was recently introduced [15]. BERT was designed based on the transformer architecture to address the constraints imposed on unidirectional approaches. Vectors generated by the BERT-based method contain rich semantic information of textual content; however, they lack topic-level features, which has been shown to improve text classification in various tasks [16].

In this study, we propose the integration of the BERT technique and latent Dirichlet allocation (LDA)-based topic modelling for classification. Some existing methods have attempted to integrate BERT with topic-based models for text analysis [17,18]. These methods use the top n words based on topics that are concatenated with the BERT model. One potential issue with this method is the shortage of topic information as a result of directly adding the top n topic words into the text, which could negatively affect the performance of a model [19]. Another possible issue with these approaches is that all the necessary input information must be compressed into a fixed-length vector, which could make it difficult to handle long texts [4]. Although this issue can be addressed by applying an average pooling operation over the vectors, not all posts in tweets contribute equally to spam identification [4]. Intuitively, it has been shown that the majority of tweet posts indicate the topic of the tweets [8]. Thus, if we can obtain the semantic information of the post, the topical word can be emphasized, which can also assist in spam identification.

Therefore, to address the aforementioned issues, this study presents a new deep learning-based method named a Topical Neural Attention Network (TNAN) for online social network spam detection that can jointly learn both the topical features and semantic information for spam detection. The proposed model has a natural symmetry between the topic and the content such that the topic representation guides content attention and the content representation guides the topic representation. To further learn the contextual features of the textual content, inspired by the recent achievements of BERT in related tasks, we use the BERT technique [15], which is a language pretraining method based on bidirectional transformers. The BERT technique can effectively learn short text information, such as that in Twitter posts, and the contextualized relationships among words in texts. By incorporating interactive neural attention, the LDA topic-based model, and the BERT technique, our model is capable of capturing the deep interaction between the global topic representation and the content representation as well as effectively learning contextualized microblog representations. After conducting a series of experiments, the results showed that the proposed model outperformed the existing approaches. The main contributions of this study are summarized as follows:

- We designed an interactive attention neural network technique that can handle topic-guided content attention and content-guided topic attention simultaneously.
- We developed an enhanced spam detection approach by concatenating contextual information from the BERT technique and attentive topic-based features from interactive neural attention. To the best of our knowledge, this is the first study to combine both content and topic attention with a neural network simultaneously for malicious spam detection.
- We examined the influence of the different components of the proposed method through ablation analysis and discovered that the model can be improved by combining all the model components.
- We conducted rigorous experiments using real-world datasets and evaluated the proposed model by comparing it with existing approaches in terms of precision, recall, and F1 metrics. The results showed that the proposed approach performs better than the baselines do.

The other sections of the paper are structured as follows: Section 2 reviews the existing works on spam detection, which include classical machine learning and deep neural network-based approaches. Section 3 and Section 4 describe the detailed methods used for the proposed model. Section 5 and Section 6 present the experimental study and the results of the suggested TNAN approach, respectively. The article is summarized in Section 7, which presents the major contributions and highlights potential research directions for future studies.

2. Related work

In an effort to safeguard users against malicious spam threats, several studies that attempt to categorize and identify spam on online social networks have been published recently. This section outlines the many ML-based OSN spam detection techniques currently in use, including both deep learning-based and classical ML techniques.

2.1. Machine learning approaches

In the literature, classical machine learning (ML) classifiers such as decision trees, k-nearest neighbours (KNNs), naïve Bayes (NB), SVMs, and random forests (RFs) have been widely used in spam detection [4]. These methods are used to classify spam textual content based on its characteristic attributes [1], such as the number of links and keywords [9]. Kontsewaya, et al. [20] compared and analysed various ML algorithms for spam identification. They showed how ML algorithms such as DTs, KNNs, and NB can be exploited for feature extraction to improve spam detection. Saeed, et al. [21] proposed an ensemble machine learning-based approach to obtain better accuracy for spam detection. Ahmed and Abulaish [22] used the Markov clustering (MC) approach and assigned probabilities of the nodes in the network by applying a weighted graph as the input of the model. Al-Zoubi, et al. [23,24] used the SVM algorithm to design hybrid ML-based spam detection. The results of the experiments revealed that the proposed algorithm was better than other previous ML-based approaches. Chen, et al. [25] utilized user and content-based features to evaluate different machine learning methods, including SVMs, for detecting spammers on Twitter [26]. They revealed the impact of various factors, such as the size and sampling of data, on spam detection.

Adewole, et al. [26] presented and analysed different ML classifiers, including the RF, SVM, and multilayer perceptron classifiers, for spamming activity identification on Twitter. In this model, different attributes, such as URL-based, network-based, and tweet-based features, are used. The authors utilized a ground truth dataset comprising approximately 25,000 Twitter accounts to validate the model. Martinez-Romo and Araujo [27] proposed an SVM-based approach by using feature-based language for spam message detection on different trending topics on Twitter. To assess the performance of the suggested

model, a dataset comprising trending topics and their associated tweets was used. The results indicated that the presented model performed better than the compared approaches did. Similarly, Al-Janabi, et al. [28] proposed a machine learning-based method that uses an RF algorithm to detect malicious links on Twitter content. To assess the performance of the model, the API Twitter technique is used to generate the evaluation dataset.

However, despite the good results of the above classical machine learning classifiers for spam classification, they occasionally suffer from computational complexity and domain dependence [4]. Thus, to address the existing shortcomings, several deep learning techniques have been introduced.

2.2. Deep learning-based method

With the recent achievements and popularity of artificial neural network (ANN) techniques in the fields of data mining and text analytics [29], several deep learning-based approaches have been introduced to address the issue of spam detection. For example, Ruan and Tan [30] presented a deep learning-based approach for classifying spamming activities. The proposed model uses the word embedding technique and a three-layer backpropagation neural network to extract salient features. Ma, et al. [31] introduced a deep learning approach based on gated recurrent units (GRUs) and the long short-term memory (LSTM) technique to solve the high-dimensional problem of prior methods for spam activities. Alom et al. [6] designed a deep learning-based method for enhancing the spam detection process. The authors applied a convolutional neural network (CNN) model based on word embeddings to learn semantic information about Twitter content. The method uses a combination of tweet textual content with metadata to improve spam classification. Xu, et al. [32] introduced a deep hybrid model by combining Bi-LSTM and CNN techniques to extract salient semantic features in textual content for spam identification.

A similar study in [33] proposed an ensembled deep neural network-based method for classifying social spamming activities in online social networks. The proposed approach particularly applies a multiobjective evolutionary feature selection algorithm (MOEFA) for normalizing features that are needed for spam identification. The experimental results indicated the advantage of the suggested approach over classical ML techniques such as SVM, NB, and RF. Gupta, et al. [34] developed a deep learning-based model utilizing text-based and user-based features for spam classification on Twitter data. The presented method outperforms other existing methods. In [3], a deep learning-based method was used to extract useful features from the word embedding technique to identify spamming activities on social networks. To validate the proposed model, the authors particularly applied tagged Twitter datasets.

Another study in [35] introduced a deep hybrid model by using LSTM and CNN methods to identify spamming activities on Twitter. To further improve the performance, by ensuring better word representation, the authors suggested exploiting Word-Net knowledge. Madisetty and Desarkar [36] presented a spam detection model that uses different features, including content-based, n-gram, and user-based features, for spam classification. The presented approach was compared with traditional classifiers, and the results showed that the proposed approach outperformed the baseline methods. Roys, et al. [37] presented a deep learning-based method that combines LSTM and a CNN to extract features from short message (SMS) texts for spam filtration in text messages.

Although the above approaches have demonstrated improved performance over their previous methods [4], most existing deep learning-based approaches typically rely solely on word embeddings such as word2vec [13] and Glove [14] as the main semantic features. However, owing to their intrinsic nature, word embeddings alone cannot guarantee the capture of better semantic information from textual content. Thus, to further enhance the performance of the existing deep learning-based methods recently, the BERT technique was introduced [15] for the NLP task. The BERT technique was designed based on the transformer architecture to alleviate the constraints imposed on unidirectional methods. To improve the existing BERT technique, some works have integrated BERT with topic-based models for text classification. For example, Peinelt [17] integrated topic information with BERT to solve domain-specific similarity problems and showed the advantage of integrating topic features with the BERT technique for better performance. A similar approach was presented in [19] by introducing an approach that vectorizes sentences corresponding to topics as extended features for classification. However, this method utilizes vectorized topic words directly, and these words generally result in external noise, which could affect model learning. To address the problem of feature sparseness, a topic-based model is combined with the BERT model to generate the top n-topic word probabilities of topics [18]. Although these methods achieve impressive performance, directly utilizing the top n words under the topic in the textual content could degrade topic information and consequently affect model accuracy [19]. Another possible issue with these approaches is that all the necessary input information must be compressed into a fixed-length vector, which could make it difficult to handle long texts [8]. Although this issue can be addressed by applying an average pooling process over the vectors, not all posts in tweets contribute equally to spam identification [4]. As a result, this issue is more challenging. Thus, unlike the existing methods [17,18], our proposed model essentially utilized attentive topic features based on the interactive attention network. To better identify the advantages of the proposed model in Table 1, we present the drawbacks of the most relevant existing methods. Additionally, to identify the novelty of the proposed model, in Table 2, we compare our method with existing methods. As shown in Table 2, most of the existing approaches essentially utilize deep learning models, more specifically, LSTM. However, very few relevant approaches have used topic modelling and the BERT model. However, none of the existing approaches utilize interactive attention schemes; this makes our proposed method novel and entirely different from the existing approaches.

3. Preliminary

This section presents the background of several important concepts used for designing the proposed model. We describe the concept of the BERT technique, followed by the classical LDA scheme, and the LSTM model is described. Details of the concepts are given in the following subsection.

Table 1
Analysis of some of the most relevant works.

Model	Description	Drawback
[30]	Uses Word embedding (WE) to extract salient features	Disregards topical and attentive features
[6]	Combines Twitter content with metadata	Disregards topical and Attentive features
[32]	Integrates Bi-LSTM and CNN to extract salient semantic features	Cannot learn attentive features
[33]	Uses a multiobjective evolutionary features selection algorithm	Cannot learn contextual features
[34]	Utilizes both text-based and user-based features	Disregards topic-aware information
[15]	Utilizes LSTM and CNN	Cannot learn attentive features
[17]	Integrates topic information with BERT	Cannot learn topic and semantic features simultaneously
[19]	Vectorizes sentences corresponding to topics as extended features	Cannot contextual features
[18]	Combines topic features with the BERT model	Cannot learn attentive features

Table 2
Comparison of some of the most relevant works with our proposed model.

Model	LSTM/DL	Topic-Aware	BERT-Integrated	Interactive Attention
[30]	√	*	*	*
[6]	√	*	*	*
[32]	√	*	*	*
[33]	√	*	*	*
[34]	√	*	*	*
[15]	√	*	*	*
[14]	√	*	*	*
[17]	√	*	√	*
[19]	*	√	*	*
[18]	√	*	√	*
TNAN	√	√	√	√

3.1. BERT technique

BERT is a method for encoding unprocessed sentences into vectors that depend on the context. Transformers are regarded as essential components of the BERT model. Transformers replace conventional encoder–decoder-based LSTM devices. The BERT technique uses next-sentence prediction (NSP) tasks and a masking language model (MLM) to enhance semantic representation learning. BERT has fine-tuning transfer learning and powerful feature extraction abilities, which makes it stand out in several NLP tasks [15,42,43]. There are two versions of the BERT technique, namely, the BERT-BASE model and the BERT-LARGE model. In this study, similar to other relevant works [43, 44], we utilize the BERT-BASE technique as our foundation model. BERT-BASE essentially consists of an encoder with 12 blocks of the transformer, 12 self-attention heads, and a hidden representation size of 768. BERT uses "word pieces" instead of "words" as input and generates a representation of the sequence as the output. In addition to general word pieces, BERT uses other special tokens, namely, [*CLS*] and [*SEP*], which are used to specify the beginning of a sentence and the end of a sentence, respectively.

In general, the transformer encoder of the BERT technique connects multihead self-attention and feedforward via a residual network structure. The multiheaded method applies multiple linear transformations to the input vector to generate different linear values. The attention weight can be computed as follows:

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

where Q, K, and V represent the query, key, and value matrices, respectively. Hence, the multihead self-attention can be expressed as:

$$MultiHead(Q,\ K,\ V) = Concatenate(head_1\ , head_2\ \ldots . head_h\ )W^o$$

$$head_f\ \ = Atten(QW_f^Q, KW_f^K, VW_f^V)$$

The projection is the parameter matrices $W_f^Q \in \mathbb{R}^{\ d_{model}\times\ d_q}$, $W_f^k \in \mathbb{R}^{\ d_{model}\times\ d_k}$, $W_f^v \in \mathbb{R}^{\ d_{model}\times\ d_v}$ and $W^o \in \mathbb{R}^{\ hd_v\times\ d_{model}}$. The output of the BERT technique varies slightly on various downstream tasks. Fig. 1 shows a visual representation of the BERT model. In Fig. 1, E_i denotes the embedding representation, *Trm* represents the intermediate representations of the same token, and T_i represents the final output.

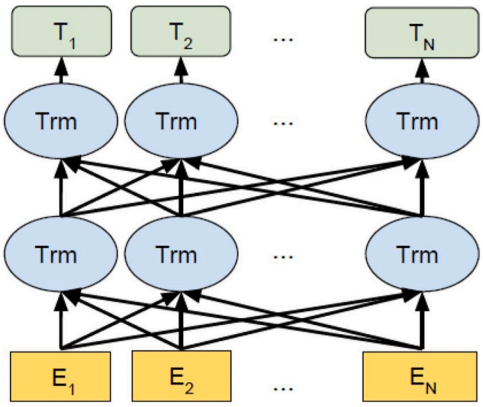

Fig. 1. Visual Representation of the BERT Technique.

3.2. LDA

LDA is a generative probabilistic technique for a corpus. The main idea is that documents are represented as random mixtures over latent topics, and each topic is characterized by a distribution over a word. LDA has shown good performance in modelling the topic of a document. However, traditional LDA cannot optimally generate topic modelling in short texts such as Twitter. Therefore, a previous study introduced the Twitter LDA technique for addressing the issue of short text. Unlike the classical LDA [39] settings in which each word has a topic label, in the Twitter LDA-based technique, a particular microblog post is likely associated with one topic. Fig. 2 graphically depicts the LDA model.

Assume that there are T topics and that each topic can be characterized by a word distribution. Let π be a Bernoulli distribution that directs the choice between background words and topic words. $\varnothing^t$ and $\varnothing^B$ represent the distributions of words for topic t and background words, respectively. To generate a text post, a topic is selected, and a bag of words is then chosen one by one according to the selected topics. The Gibbs sampling method [40] can be used to estimate the Twitter LDA parameters. Each text s can be assigned a topic z, and then the M most likely words can be extracted as the semantic information of topic z.

3.3. LSTM

Since text is sequential data, even slight alterations in word order can have an impact on the overall meaning of a sentence. Nevertheless, the word dependency of context cannot be easily extracted by conventional feedforward neural networks. To extract sequential and contextual properties from these data, researchers have created sequential models such as recurrent neural networks (RNNs). There are three layers in an

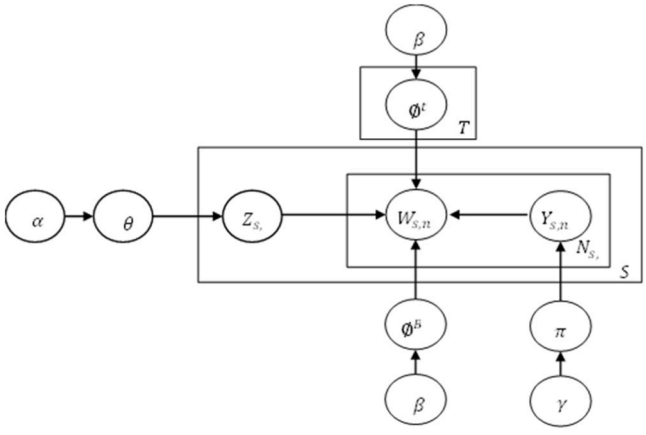

Fig. 2. Graphical representation of the LDA model.

RNN: the input layer, the hidden layer, and the output layer. However, gradient explosion and disappearance become more apparent during the training process as the sentence length increases. To address this issue, long short-term memory (LSTM) introduces a cell state to store long-term memory. Consider the text sequence $X = (x_1, x_2 \ldots\ldots\ldots x_N)$. For each position x_t, the transition of the LSTM can be computed as follows:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$$

$$f_t = \sigma(W_f x_t + \quad U_f h_{t-1} + b_f)$$

$$O_t = \sigma(W_o x_t + \quad U_o h_{t-1} + b_o) \tag{1}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_c x_t + U_c h_{t-1} + b_c)$$

$$h_t = O_t \odot \tanh(c_t)$$

where h_{t-1} and the cell state c_{t-1} represent the previous output layer and the cell state, respectively. i_t, f_t, and o_t denote the input gate, forget gate, and output gate, respectively. c_t and h_t are the cell state and next output, respectively. σ and$\odot$ are sigmoid functions and elementwise multiplications, respectively. $W \in \mathbb{R}^{d_{hidden} \times d_{hidden}}$ and$U \in \mathbb{R}^{d_{hidden} \times 1}$ represent the weight matrices to be learned. $b_c \in \mathbb{R}^{d_{hidden}}$ is a bias vector. Fig. 3 graphically depicts the LSTM.

4. Proposed method

In this section, we present the detailed methodology of the suggested model. First, we describe the proposed model, which comprises different components. Second, we describe different parts of the model, which include the sequence encoder, interactive attention with topic modelling, the BERT technique, and the prediction layer.

4.1. Overview of the proposed model

The main purpose of this research is to design an enhanced deep learning-based method for spam detection on social networks. Motivated by the finding that legitimate users and spammers generally focus on various trending topics and use different words in their textual content accordingly, we extract attentive topic features, which can be important tools for distinguishing between textual content written by spammers and legitimate users. We design an interactive neural attention model that can capture both content and topic word information simultaneously. To better learn the semantic information from the tweet post, we use the Bi-LSTM encoder model, which is very powerful in learning sequential information of texts. To achieve topic modelling of the word distribution, we utilize a Twitter LDA, which is very effective in short text analysis [38]. Thus, we can obtain content and global topic attentive features. To further learn the contextualized features of the textual content, inspired by the recent achievements of BERT in related tasks, we use the BERT technique [15], which is a language pretraining method based on bidirectional transformers. The BERT technique can effectively learn the semantic information of short text and the contextualized relationships among the words in texts. The generated attentive features are then integrated with contextualized embedding vectors from the BERT technique to obtain the final rich feature vectors, which are then passed to the SoftMax function for the final spam detection. In this way, the proposed model can learn the deep interactions of both the local content representations and the global topic representations, as well as the contextualized semantic features of a textual post. The proposed model is composed of four parts: 1) A sequence encoder layer that uses a Bi-LSTM model to capture the semantic and sequential information of the text. 2) An interactive neural attention mechanism learns the attentive features by jointly learning the topic word features and the local content word features simultaneously. 3) The BERT technique is used to learn the contextualized features of the textual content from microblogs. 4) The prediction layer, which is the output layer that uses the SoftMax function to provide the final classification of the spam. Fig. 4 shows the general framework of the TNAN approach. The proposed method is shown as follows:

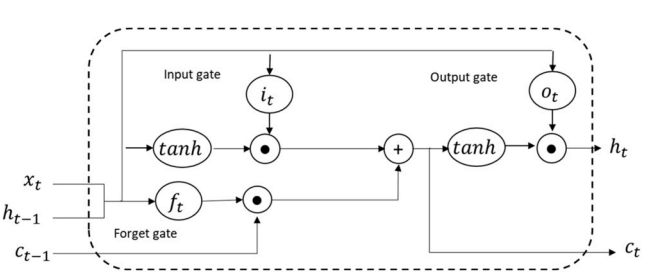

Fig. 3. Graphical representation of LSTM.

4.1.1. Sequence encoding layer

To represent words, we embed each word into a low-dimensional vector $L_W \in \mathbb{R}^{d_{emb} \times |V|}$, where d_{emb} is the embedding dimension and $|V|$ is the number of words in the vocabulary. To better understand the semantic associations of the textual content, we use a pretraining technique. Given input textual content, the embedding $X_t \in \mathbb{R}^{d_{emb} \times 1}$ is used for each word in the textual content to generate a word embedding (WE). Thus, a text of length N can be denoted with the sequence of vectors $X = (x_1, x_2 \ldots\ldots\ldots x_N)$.

Next, we use the LSTM technique, which is a variant of the recurrent neural network (RNN) technique and has been shown to be good at modelling sequential information. Controlling the data passage along the sequences improves long-range dependencies. LSTM uses different gates, which include the input gate, output gate, and forget gate. To enable hidden state capture of both future and previous contextual data, we used a Bi-LSTM technique, which is an extended version of the LSTM. The Bi-LSTM technique uses semantic data from both directions (forwards and backwards).

For a text sequence $X = (x_1, x_2 \ldots\ldots\ldots x_N)$, the forwards LSTM considers the sequence from x_1 to x_N, and vice versa. The backwards LSTM considers the sequence from x_N to x_1 and, equally, processes the sequence based on Eq. (1). Then, the forwards hidden state $\overrightarrow{h}_t$ and backwards hidden state $\overleftarrow{h}_t$ are combined: $\mathrm{h_t} = [\overrightarrow{h}_t; \overleftarrow{h}_t]$. Consequently, $\mathrm{h_t}$ combines all the information of the sequence in $\mathrm{x_t}$. The output of the LSTM can be obtained as a sequence of hidden states $(\mathrm{h_1}, \mathrm{h_2} \ldots \mathrm{h_t}) \in \mathbb{R}^{\mathrm{d}}$. Each annotation contains information about the whole review with a focus on the $i-th$ word. After the hidden state $h_t = \left[h_t^1, h_t^2 \ldots\ldots h_t^f\right]$ is generated, we can obtain the content feature representation P as follows:

$$P = \left[h_1^p, \quad h_2^p \ldots\ldots h_n^p\right] \tag{2}$$

where h_j^p represents the features, the vector is based on Eq. (1) for the $j-th$ hidden state, and n is the sequence length.

4.1.2. Topical embeddings

Topic modelling is effective in identifying important information in textual content. In topic modelling, the assumption is that a document is composed of different topics, each of which is a distribution with respect to the words in the vocabulary. By fitting the models, each document can be represented via learned topics, and each topic can be learned through the probability distribution of words. The main semantic information is represented by topical words. In this study, we suggest integrating information such as prior knowledge into the Bi-LSTM technique. To model the topics of the textual content, we utilize Twitter LDA [38], which is effective in topic modelling for short text analysis.

To generate a text post, a topic is selected, and a bag of words is then chosen one by one according to the selected topics. Therefore, the topic

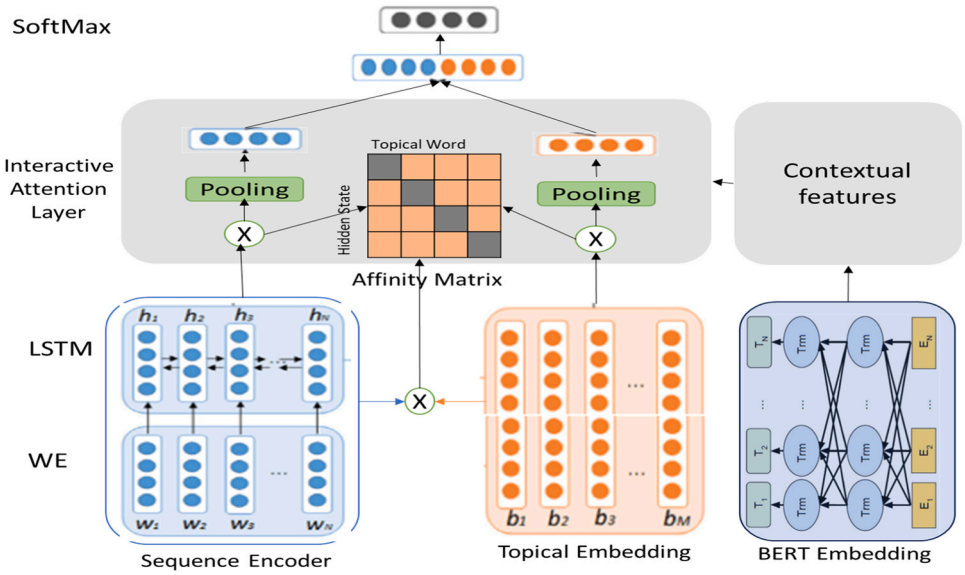

Fig. 4. General framework of the TNAN approach.

semantic information of the text can be represented by a sequence of embedding vectors of the topic word, $[b_1, b_2 \ldots\ldots\ldots b_M]$, where M is the number of topical words selected for each topic. Thus, we can obtain a topic feature vector given as $Q \in \mathbb{R}^{M \times l_1}$

4.1.3. Interactive attention

As stated previously, in tweet posts, not all words contribute equally to social spam identification. Therefore, inspired by the successful application of the neural attention technique, we propose the use of neural attention networks to automatically capture the most relevant words in microblogs for the spam detection task. We propose a topical interactive network that can model the topic and content information of microblog posts simultaneously. The main purpose of the interactive mechanism is to allow simultaneous learning of topic word representations as well as local content word representations. The idea here is to allow pairwise learning such that the information from the content word representation can directly influence the computation of the topic word representation and vice versa. Topic feature representation is utilized as a basis when learning content attention, and content features are used as the basis when learning topic attention. To achieve this, first, we need to compute the affinity matrix $\overline{C}$ [41] based on the topic and content feature representations as follows:

$$\overline{C} = \tanh\left(P^T \quad W_{\overline{c}} Q\right) \tag{3}$$

where $P \in \mathbb{R}^{N \times l_1}$ denotes the content feature representation, $Q \in \mathbb{R}^{M \times l_1}$ represents topic feature representations, $W_{\overline{c}} \in \mathbb{R}^{l_1 \times l_1}$ represents a weight to be learned, and tanh represents nonlinearity. Each entry in $\overline{C} \in \mathbb{R}^{N \times M}$ represents the affinity between the respective pair of content and topic representations. Second, having obtained the matrix, similar to the study in [41], the matrix is used as a feature to determine the topic and content feature attention by applying a single perceptron layer as follows:

$$\beta_p \quad = \tanh(P \quad W_y + \quad \overline{C}(QW_z), \quad a^P = softmax\left(\beta_P V_y\right) \tag{4}$$

$$\beta_q \quad = \tanh(Q \quad W_z + \quad \overline{C}(PW_y), \quad a^q = softmax\left(\quad \beta_q V_z\right) \tag{5}$$

where W_y, $W_z \in \mathbb{R}^{l_1 \times \quad l_2}$ and V_y , $V_z \in \mathbb{R}^{l_2}$ are the weights and $a^p \in \mathbb{R}^N$ and $a^q \in \mathbb{R}^M$ represent the topical feature attention probabilities and content feature attention probabilities, respectively. Last, the attention vectors for the topic word and local content word features can be given as the weighted sum as follows:

$$\widetilde{P} = \sum_{i=1}^{M} \quad a_M^q \quad q_M \quad , \quad \widetilde{Q} \quad = \sum_{j=1}^{N} \quad a_N^q \quad Q_N \tag{6}$$

Next, the resultant vectors derived from Eqs. (3) – (6) are normalized with pooling to form a fixed-length vector V_{vec}. Thus, we have:

$$V_{vec} = [\text{et al}] \tag{7}$$

4.1.4. BERT embedding

The main purpose of the BERT embeddings is to enable the model to learn the contextualized features in addition to the attentive topical features for better classification. Thus, to further learn the contextual features of the textual content, inspired by the recent achievements of BERT in various related tasks [15,42,43], we use the BERT technique [15], which is a language pretraining method based on bidirectional transformers. The BERT technique can effectively learn the semantic information in short text information, such as in Twitter posts, and effectively learn the contextual relationships among words. For each sentence, the BERT method generates a final vector of length 768, which can be represented as:

$$V_{BERT} = [T_1 . T_2 \ldots\ldots\ldots T_L] \tag{8}$$

where L is the length of the vector, which is 768 in this case.

4.1.5. Prediction layer

This is the stage where the final spam prediction is achieved. After the attentive topic feature vector in Eq. (8) and the contextualized semantic vectors from the BERT technique are obtained, they are combined. The contextualized semantic word vector matrix V_{BERT} in Eq. (8) and the attentive topic vector V_{vec} from the interactive attention network in Eq. (7) are integrated to generate a final vector, T_{B+L}, which is used as the input to the prediction layer and then subjected to a nonlinear layer and a shared space of the targeted C classes, as given in Eq. (9) below. The probability of classifying the textual content as spam or nonsam is computed via Eq. (11).

$$T_{B+L} = \quad V_{vec} + V_{BERT} \tag{9}$$

$$c_i = \tanh(W'_{B+L} \quad + T_{B+L} + b_{B+L}) \tag{10}$$

$$Softmax(\quad c_i) = \frac{\exp(c_i)}{\sum_{i'}^{C} \exp(c_{i'})} \tag{11}$$

where W'_i and b_i are the weight matrix and bias, respectively.

5. Experimental analysis

To validate the proposed method, we conducted a series of experiments that are described in the following subheadings. In this section, we present the datasets used in the experiments and the baseline approaches for comparison. Next, performance evaluation measures and experimental results are presented and discussed. The experiment is specifically designed to address the following experimental questions (EQs):

EQ1: Can the proposed model outperform the existing methods for social spam detection?
EQ2: What is the sensitivity of the proposed TNAN parameters, such as epoch and batch size?
EQ3: What is the influence of the different components of the TNAN models?

5.1. Dataset

To assess the performance of the suggested TNAN approach, we need a set of labelled tweet datasets. Thus, we use two popular datasets, which are described below.

1. Microblog PCU [32] (Dataset 1): This dataset was obtained from the UCI database site. The dataset contains basic characteristic attributes of users, which include the number of fans, users, gender, number of followers, and content posted by the users. For this study, a portion of the posted content data was extracted for our experiments. There are a total of 2000 data points, which include 400 spam and 1600 nonspam.
2. Honeypot datasets (Dataset 2) [45]: This dataset, which comprises 41499 users (19276 spammers and 22223 nonspammers), was collected over nine months. The dataset has six different files, which include content polluter files, content polluter tweeting files, legitimate user files, legitimate user tweeting files, legitimate user following files, and content polluter following files. For our experiment in this study, two files, the tweet files of legitimate users and the tweet files of content polluters, are used since these are the only files containing posts of spammers and nonspammers.

5.2. Preprocessing

Before the datasets are passed into the network for the experiment, some preprocessing tasks must be conducted on the dataset. Following the NLTK standard for preprocessing, all the datasets are initially transformed into lowercase and then split into separate sentences. All special characters, stop words, alphanumeric characters, emoticons, symbols, unknown characters, etc., are filtered accordingly. To obtain normalized datasets, null values are also removed from all the columns of the data. The dataset is divided at a ratio of 20:80 % for testing and training. We applied TweetTokenizer for topic features and a tokenizer, particularly for the BERT inputs. As previously noted, the BERT technique uses special tokens [*CLS*] and [*SEP*] to effectively understand the inputs. [*CLS*] is added at the beginning of a single sentence, and [*SEP*] is added at the end of a single sentence as a separator between the sentences.

5.3. Experimental settings

The experiment was conducted via the Python programming language based on the Keras tool and trained on Windows 10 with 384 GB memory and an NVIDIA RTX A6000 GPU. Like other related approaches [16], we employ BERT-BASE, which uses an encoder with 12 attention heads, 12 transformer layers, and a hidden size of 768. Unless otherwise stated, the proposed model and the baselines were trained with 50 words as the sequence length. For the LSTM model, 500 and 300 are used as the hidden state and word embedding dimensions, respectively. We use a minibatch stochastic gradient descent method and the Adam algorithm to train the models [46]. The hyperparameter β1 is set to 0.999, and β2 is set to 0.9 for optimization. 0.001 is used as the learning rate. For our TNAN approach, after testing with different topic sizes T and different numbers of topical words M, 300 and 40 were determined to be the best settings for T and M, respectively. All the values were selected via the grid search technique.

5.4. Baselines

To validate the performance of the TNAN model, we identified different baseline methods for comparison with the proposed approach. The model is compared with classical spam detection methods that use handcrafted features and then with related deep neural network-based methods that use raw data as inputs for spam detection. These models are employed for comparison because of their close relationship with the proposed model. The models are also selected based on their outstanding performance compared with their relevant state-of-the-art counterparts. The baselines used for the comparison are explained as follows:

- SVM: In this approach, the SVM algorithm is used as a classifier, which is trained and applied for spam classification.
- Matrix factorization method (MF) [47]: This approach uses the matrix factorization (MF) technique, which exploits a social relationship graph and labelled data for spam classification.
- Deep-learnt features (DLFs) [7]: This model is based on word2vec and the Bi-LSTM to learn embeddings and user representations for the classification of spam messages.
- Twitter Spam Detection based on Deep Learning (TSD) [3]: This model uses a paragraph vector modelling method for learning a tweet-level representation by averaging all the tweets posted by the same users.
- DeepSBD [8]: This is a deep learning approach with an attention mechanism for social bot detection (DeepSBD) that uses a convolution structure based on the Glove vectors for spam classification.
- DeepTwitter [6]: This is a type of Twitter spam identification that uses a deep neural network based on a convolutional network for learning the representation of each tweet for spam prediction.
- Bi-LSTM [32]: This is a deep learning-based method that uses a self-attention Bi-LSTM model integrated with a lightweight BERT and ALBERT for spam detection

5.5. Evaluation metrics

To assess the proposed TNAN approach, we use precision (Pre), recall (Rec), F1 scores, and accuracy (Acc) as the evaluation tools (metrics). These metrics can be represented in terms of the true positive (TP), which represents the number of spam tweets that are labelled as spam. False positive (FP) represents the number of nonspam tweets that are wrongly classified as spam. True negative (TN) represents the number of nonspam tweets that are correctly considered nonspam, and false negative (FN) refers to the number of spam tweets mistakenly classified as spam. These metrics can be represented as follows:

Recall: This can be defined as the ratio of correctly classified spam to total spam. It can be expressed as:

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

F1 score: This is the harmonic mean of the recall and precision, which can be expressed as:

$$F1score = \frac{2 * precision * Recall}{Precision + Recall} \tag{13}$$

Precision: This metric can be defined as the ratio of correctly classified spam to total actual spam. It can be given as:

$$precision = \frac{TP}{TP + FP} \tag{14}$$

Accuracy: This is the ratio of the predicted values to the total predictions. It is expressed as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{15}$$

6. Results and discussion

In this section, the experimental results of the proposed method are presented and discussed. Table 3 and Table 4 show the performance comparison of the TNAN model with the baselines on Datasets 1 and 2 in terms of the Rec, Pre, and F1 scores and Acc, which proves the superiority of the TNAN model over the baselines. The results show that our proposed method performs significantly better than the baselines do.

Table 3 (using dataset 1) shows that our proposed method significantly outperforms the baselines, with average gains of 17.54 %, 6.19 %, 11.91 %, and 12.27 % in terms of Rec, Pre, F1, and Acc, respectively. Additionally, the findings in Table 4 (using Dataset 2) show that our proposed model improved, with gains of 11.81 %, 4.38 %, 8.12, and 7.42 in terms of Rec, Pre, F1, and Acc, respectively.

To investigate the improvement of our proposed model over the baseline methods, we also run statistical significance tests (t tests). A statistical t test was conducted via SPSS software. A t test was used to determine if there was a significant difference between the means of the two samples. To compare the means of two findings that reflect the test groups, we used a two-tailed paired t test procedure. There is a significant difference between the two outcomes when the t test result has a low significance value, usually less than 0.05. This finding indicates a considerable difference in the outcomes between the two groups, with less than a 0.05 chance that the two results originated from the same group. Our results showed that the performance gains are statistically significant at $p < 0.05$. This finding confirms the effectiveness of the suggested TNAN method.

From the results, several observations can be made concerning the performances of the baselines and the proposed TNAN model. First, the performance of the SVM-based model, which is a popular traditional machine learning algorithm, is relatively worse than that of the other benchmark approaches in terms of Rec, Pre, F1, and Acc. However, the MF model, which essentially uses matrix factorization and exploits extra features for learning latent factors, performs better than the other feature-based approaches, namely, SVMs. Tables 3 and 4 show that all the models that use only features for the classification task (SVMs and MF) generally performed worse than did the deep learning-based approaches DLF, TSD, DeepTwitter, and DeepSBD, which generally use word embedding techniques for feature extraction. This finding clearly shows the suitability of deep learning-based methods for spam classification, which justifies the previous findings.

Table 3
Performance comparison of the TNAN model on Dataset 1.

Model	Rec	Pre	F1	Acc
SVM	0.672	0.821	0.739	0.748
MF [47]	0.769	0.91	0.834	0.825
TSD [3]	0.795	0.88	0.835	0.842
DLF [7]	0.826	0.871	0.848	0.855
DeepTwitter [6]	0.837	0.851	0.844	0.837
DeepSBD [8]	0.873	0.882	0.877	0.884
Bi-LSTM [32]	0.891	0.912	0.901	-
TNAN	0.914	0.927	0.920	0.927

Table 4
Performance comparison of the TNAN model on Dataset 2.

Model	Rec	Pre	F1	Acc
SVM	0.765	0.844	0.803	0.815
MF [47]	0.782	0.922	0.846	0.858
TSD [3]:	0.808	0.894	0.849	0.861
DLF [7]	0.839	0.885	0.862	0.874
DeepTwitter [6]	0.85	0.865	0.858	0.870
DeepSBD [8]	0.886	0.896	0.891	0.903
TNAN	0.927	0.941	0.934	0.946

Among all the deep-neural network-based models, TSD, which uses a traditional method of user representation by considering each user in the textual content as independent of others, shows lower performance than other deep learning methods, namely, DeepTwitter, DLF, and DeepSBD. DeepTwitter uses convolutional structures, which are good for feature extraction, and Deep-learnt features uses Bi-LSTM to learn the inner relationships among words within each text. Thus, they can perform better than the TSD. However, DeepSBD, which uses a CNN and glove model as well as an attention network to extract feature representations, achieves the best performance among all the baseline approaches. More importantly, in all the cases, our proposed TNAN model outperformed all the baselines. This finding shows the power of our proposed model in terms of feature learning for better classification; this can be combined with the introduction of attentive topic modelling based on Twitter LDA, which is integrated with the BERT technique.

The results of our experiments showed that our proposed model outperformed popular traditional machine learning algorithms such as SVM and MF. Additionally, the results showed that our proposed model performed better than the existing feature-based deep learning models, including TSD, which utilized a traditional method of user representation by considering each user in the textual content. Moreover, even the state-of-the-art deep learning-based methods for spam detection, such as DeepTwitter, DLF, and DeepSBD, still outperform our proposed model, with significant improvements.

6.1. Ablation results

To further assess the credibility of our TNAN approach in terms of the influence of the various components, an ablation study is carried out. Four different versions of the model were developed. These different versions were further assessed by comparing them with the standard setting of the proposed approach (TNAN -Standard). In this case, the TNAN-Standard is the default setting of the model, which comprises all the model components, as described in Section 3. The different versions of the proposed TNAN model are explained as follows:

- TNAN-Topic: In this setting, only the topic modelling generated from the Twitter LDA model is used for the final spam classification. Thus, in this version of the model, the interactive attention and the LSTM sequence encoder parts are not considered. This setting is essentially used to examine the impact of interactive attention and integrate the BERT technique into the model performance.
- TNAN Standard without BERT: This is the standard setting of the proposed approach, which comprises all the default components of the model without BERT.
- TNAN-Content: In this setting of the model, the topical word distribution, and the interactive attention mechanism, are disregarded, and only the LSTM encoder based on pretrained embedding is used. This version is used to examine the impact of topic integration and the BERT technique on model performance.

- TNAN-Content-BERT: In this setting of the model, both the LSTM encoder and the BERT model are considered in the model training, whereas the topic modelling based on the Twitter LDA is disregarded. This setting is used to examine the influence of integrating topic modelling on model performance.
- TNAN-Topic-BERT: In this version of the model, only the LSTM sequence encoder based on pretrained word embedding is disregarded, while other components, namely, topic modelling and the BERT technique, are used in model training. This setting is used to examine the impact of the LSTM encoder as well as the interactive attention on the model performance.

Figs. 5 and 6 show the experimental results of the ablation study, indicating the performances of the different variants of the model. To better examine the data sparseness and simulate the real social network environment, in the ablation study, we set the ratio of the number of spam to the number of nonspam to 5050 and 20:80 percent spam and nonspam, respectively. The results indicate that TNAN-Topic-BERT essentially outperforms all the other variants, namely, TNAN-Topic, TNAN-Content, the TNAN standard without BERT and TNAN-Content-BERT, in all the cases. This finding shows the influence of the LSTM encoder in contrast to topic modelling, which uses topical word representation. However, the results in Fig. 5(a-b) and 6 (a-b) indicate that the TNAN-Standard model, which is the standard version of the model, outperforms all the other variants in all the cases across all the datasets. This finding significantly explains the impact of integrating the BERT technique with the interactive attention mechanism utilizing topic modelling for spam classification. The results of the standard version signify the influence of combining all the main components of the model, namely, LSTM, topic modelling, and the BERT technique, on model performance.

6.2. *Parameter sensitivity*

For further evaluation, we examine the influence of different parameter settings on the performance of the proposed method. Notably, the best and most important hyperparameters for the proposed TNAN model were obtained via an extensive grid search, which is one of the best tuning approaches for model hyperparameters. The main hyperparameters, which significantly influence model performance, include the number of epochs, learning rate, batch size, and sequence length. Notably, to better examine other hyperparameters, the learning rate is fixed based on the basic setting described in the previous section. In the following subsections, the influence of the proposed model

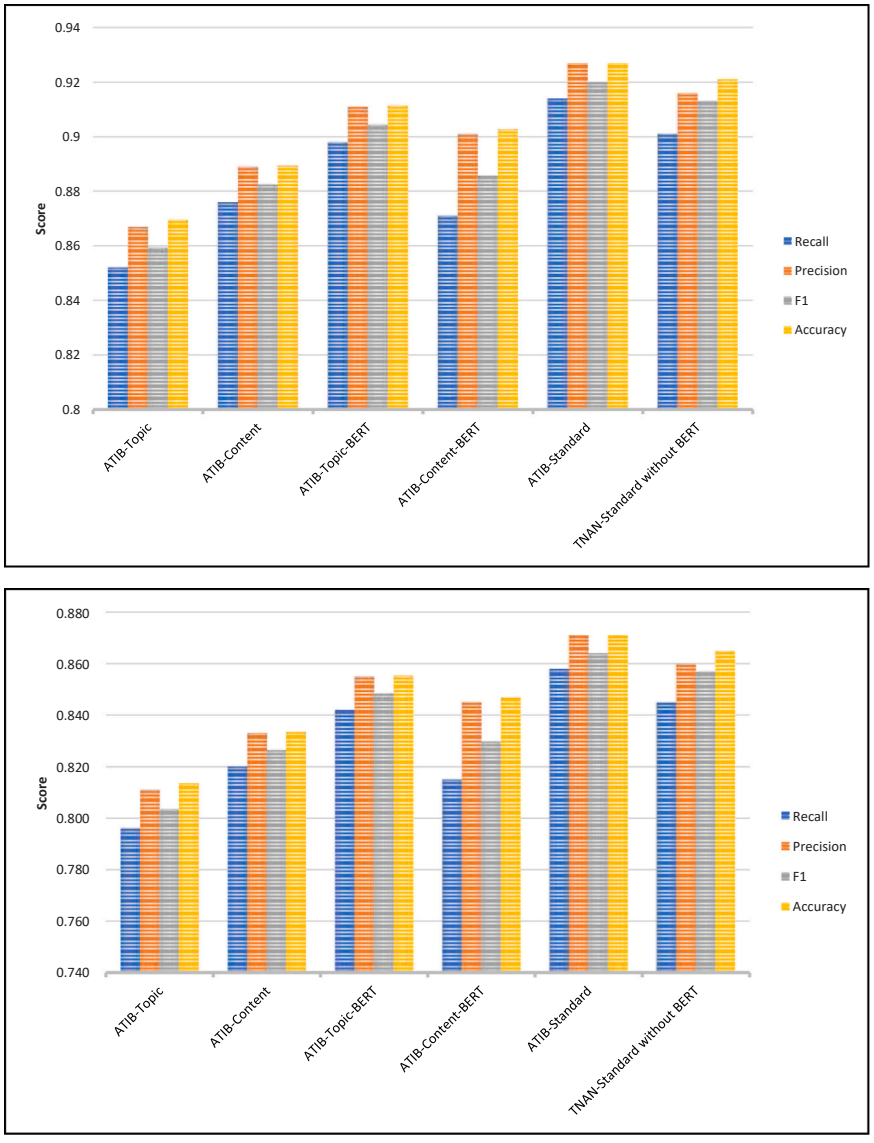

Fig. 5. (a). Ablation results on dataset 1 (20:80). Fig. 5(b). Ablation results on dataset 1 (50:50).

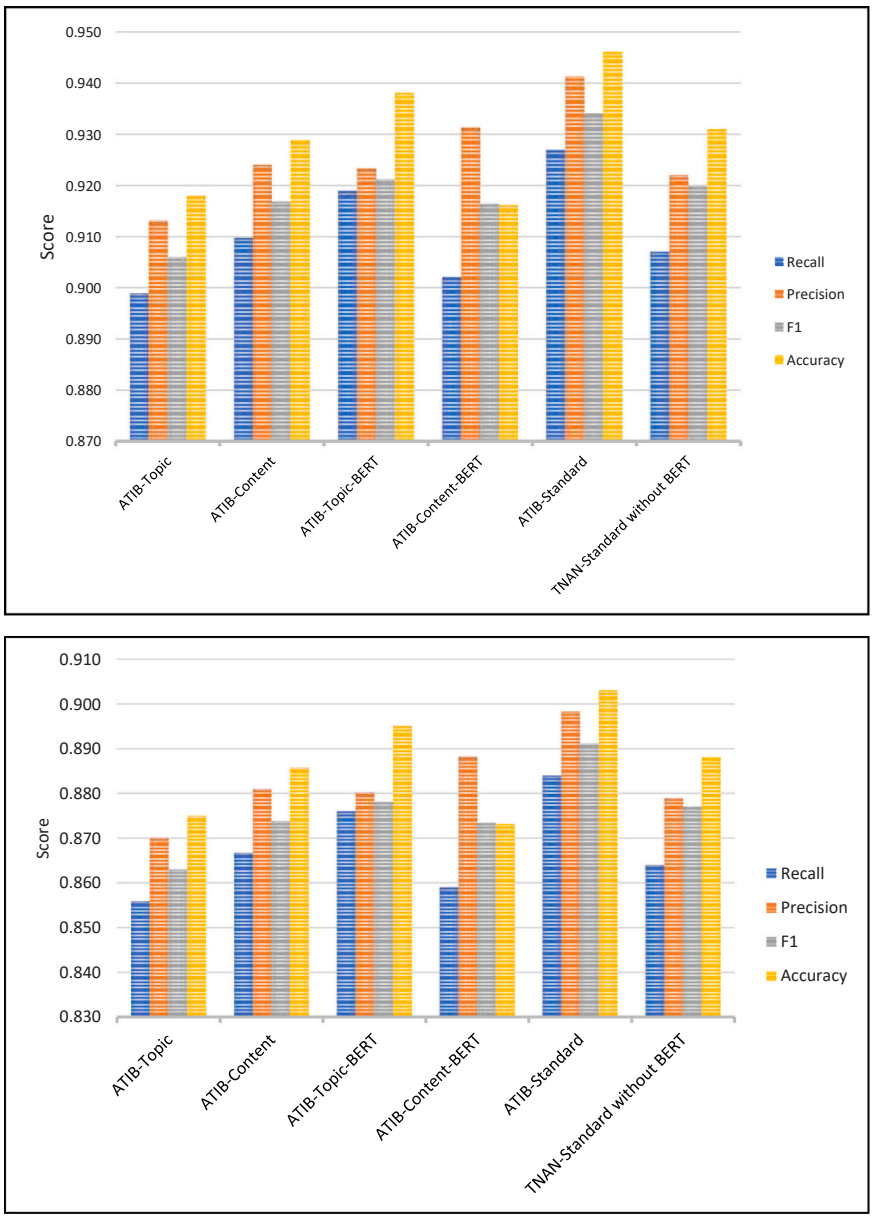

Fig. 6. (a). Ablation results on dataset 2 (20:80). Fig. 6(b). Ablation results on dataset 2 (50:50).

parameters across all the datasets is presented. Therefore, based on the grid search technique, three parameters, namely, the number of epochs, batch size, and sequence length, are examined as selected.

6.2.1. *Impact of the epoch*

The number of epochs is the number of times that models can work across the training set. One important parameter that can significantly impact the performance of the training model is the number of epochs. An epoch means training the model with all the training sets per cycle. In many cases, the accuracy score improvement of a model is directly proportional to the gradually increasing number of epochs. However, the issue of overfitting always arises when the number of epochs is very large. Thus, it is essential to search for the best and most influential value. Fig. 7(a-b) shows the influence of the number of epochs on the performance of the proposed TNAN model.

Fig. 7(a-b) shows that as the number of epochs increases, the proposed model performance also gradually improves slightly, and after reaching some point, the performance begins to decrease. Thus, the classification of the suggested model improves slightly with the number of epochs. As shown in all the figures, the best accuracy is achieved when the number of epochs is 100. The experimental evaluation also indicated that when the number of epochs is 120, the accuracy of the proposed model decreases, which means that it stops training. Consequently, a value of 100 is used as an ideal setting for the epoch parameter.

6.2.2. *Impact of sequence length*

The sequence length can be defined as the number of input tokens that can be processed by the model. Sequence length is one essential parameter that impacts the BERT training process. In general, BERT can allow a sequence length of up to 512. The BERT model adopts a padding process (filling with zeros) to ensure equal length if the input sequences are shorter than the specified length. Subsequent texts must be normalized to obtain the exact value for a sentence with a length greater than the defined values. In general, the sequence length can significantly impact the accuracy of a model; therefore, selecting the most effective value for model improvement is important. For example, if the length of the sequence is too short, much semantic information from the data can

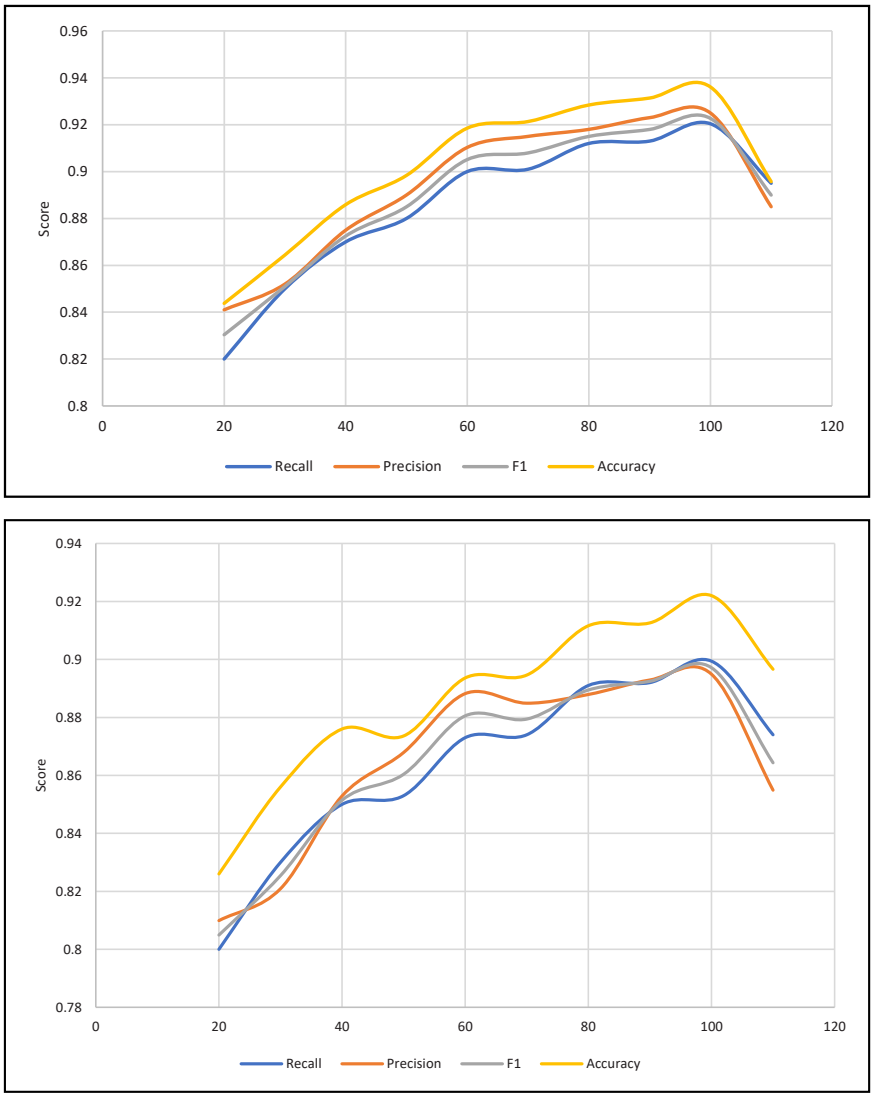

Fig. 7. (a). Impact of the number of epochs on the model performance on dataset 1. Fig. 7(b). Impact of the number of epochs on the model performance on dataset 2.

be missing. However, selecting long sequences can affect the training process of the model. Fig. 8(a-b) show the experimental results when various sequence length values are used.

Fig. 8(a-b) show that the performance of the model gradually improves as the sequence length increases. When the sequence length is 8–16, the proposed model attains the best performance. However, when the sequence length continuously increases, a memory leak is attained, and after sequence length 18, the performance of the proposed model starts to decrease. Notably, to save substantial memory, the model should be fine-tuned with a shorter maximum sequence length. Thus, the value of 16 is selected to be the ideal best sequence length.

6.2.3. *Impact of batch size*

The batch size can be referred to as the number of training samples that propagate through the network. Batch size is an important parameter that significantly affects the memory of a model and critically influences model training. In general, as the batch size increases, more memory space will be needed. However, selecting a smaller batch size can cause the learning process to fluctuate and become more complex, essentially increasing the convergence time of a model. Thus, it is essential to carefully select the best batch size to obtain a better improvement. In our experiment, after other hyperparameters are fixed, the model is tested with a batch size varying between 4 and 512. Fig. 9 (a) and 9(b) show the impact of the batch size on the model performance. As shown in the figures, when the batch size fluctuates from 32 to 128, the performance of the proposed method decreases.

When the batch size exceeds 64, the performance of the model starts to decrease. Thus, as the batch size continues to increase, the accuracy also increases. The accuracy is best when the batch size is 128. At this time, the fastest convergence of the model is attained. Therefore, 128 is selected as the batch size.

6.3. *Scalability*

To investigate the scalability of the proposed model, we conducted experiments to measure the running time of our model alongside the compared approaches. Fig. 10 shows the running time of our suggested approach in comparison with those of the baseline methods. Fig. 10 shows that our suggested approach results in a relatively greater computation time than do some of the baseline methods, namely, TSD, MF, and SVM. However, compared with DeepSBD and DeepTweeter, which also use deep learning-based approaches, our method has a relatively lower running time. Therefore, the empirical running time of our proposed method is relatively acceptable compared with those of other deep learning-based approaches.

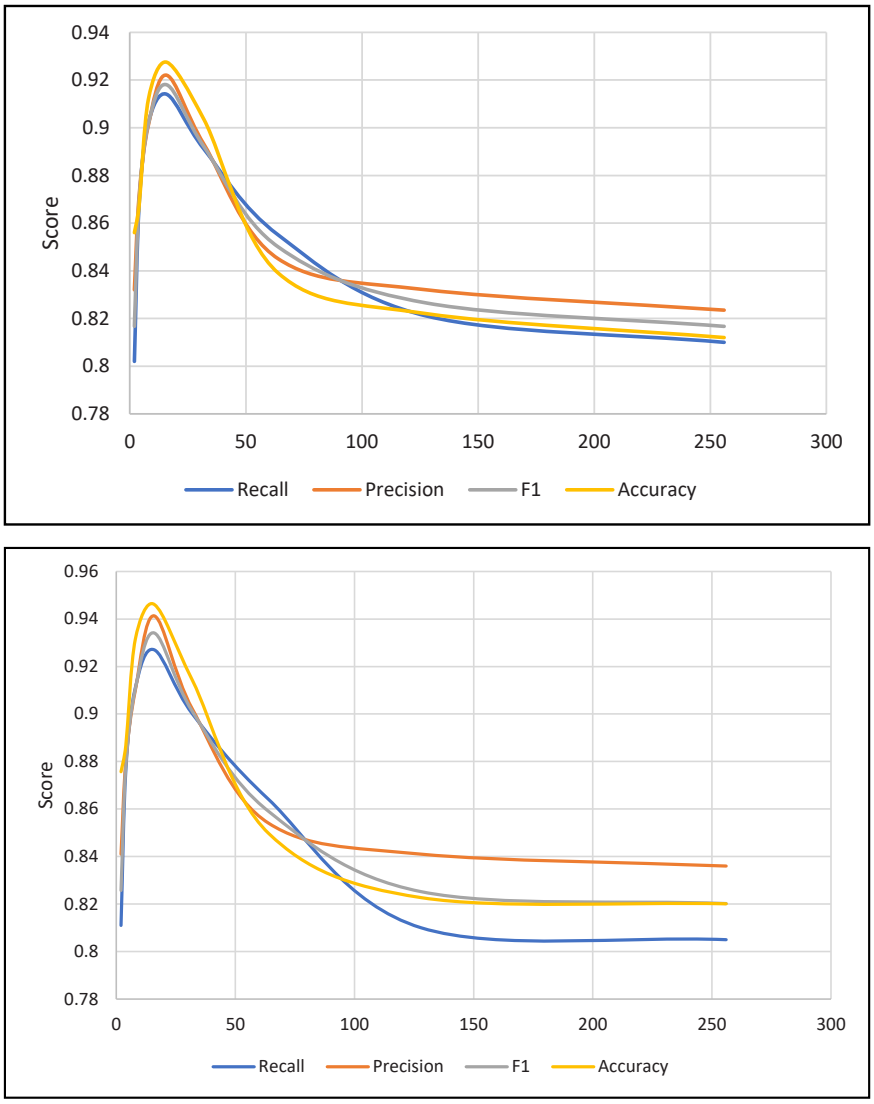

Fig. 8. (a). Impact of sequence length (SL) on model performance on dataset 1. Fig. 8(b). Impact of sequence length on model performance on dataset 2.

6.4. Qualitative analysis

In this section, we present examples demonstrating how our proposed model can learn the most informative words from user-generated text for improved spam detection. As described previously, the model employs an attention mechanism that enables it to identify and focus on the most relevant words in tweets related to spamming behaviour. To demonstrate how the model uses interactive attention to learn these key words, we randomly select samples of tweets written by both legitimate users and spammers from the honeypot dataset. Fig. 11 shows a visualization of the attention weights applied to the tweet text. Certain words are shaded in varying shades of blue. Words with the highest attention weights are coloured dark blue, those with moderately high attention scores are shaded light blue, and words with medium to low scores are not shaded.

7. Conclusion and future work

With the proliferation of online social networks such as Twitter, social threats have become a major issue, with violent messages, malware, and malicious links spreading across these platforms. With recent advancements in artificial intelligence, deep neural networks have been applied to identify spam in online social networks. However, most current deep learning-based methods generally rely on traditional word embedding techniques, which fail to address spam identification effectively. Therefore, to address the issue of spam detection more effectively, this paper proposes an enhanced deep learning-based method that integrates the BERT technique with a topic model for online spam detection. The proposed model comprises four main components: the sequence encoder layer, which uses Bi-LSTM to capture the semantic and sequential information of the text; the interactive neural attention mechanism, which learns attentive features by jointly considering topic word features and local content word features simultaneously; the BERT technique, which is used to learn contextualized features of textual content from microblogs; and the prediction layer, which uses the SoftMax function to provide the final classification of spam. The proposed TNAN approach was evaluated on real-world datasets, and the results revealed that our model outperforms the baselines in terms of recall, precision, and accuracy. One important future research direction is to consider enhanced transfer learning models, such as attentive CNNs and generative models, as classifiers based on contextualized topic features. Another future direction is to investigate how the issue of ambiguity can be addressed in both topic and contextualized feature extraction, given that many words are ambiguous and some words are independent of context.

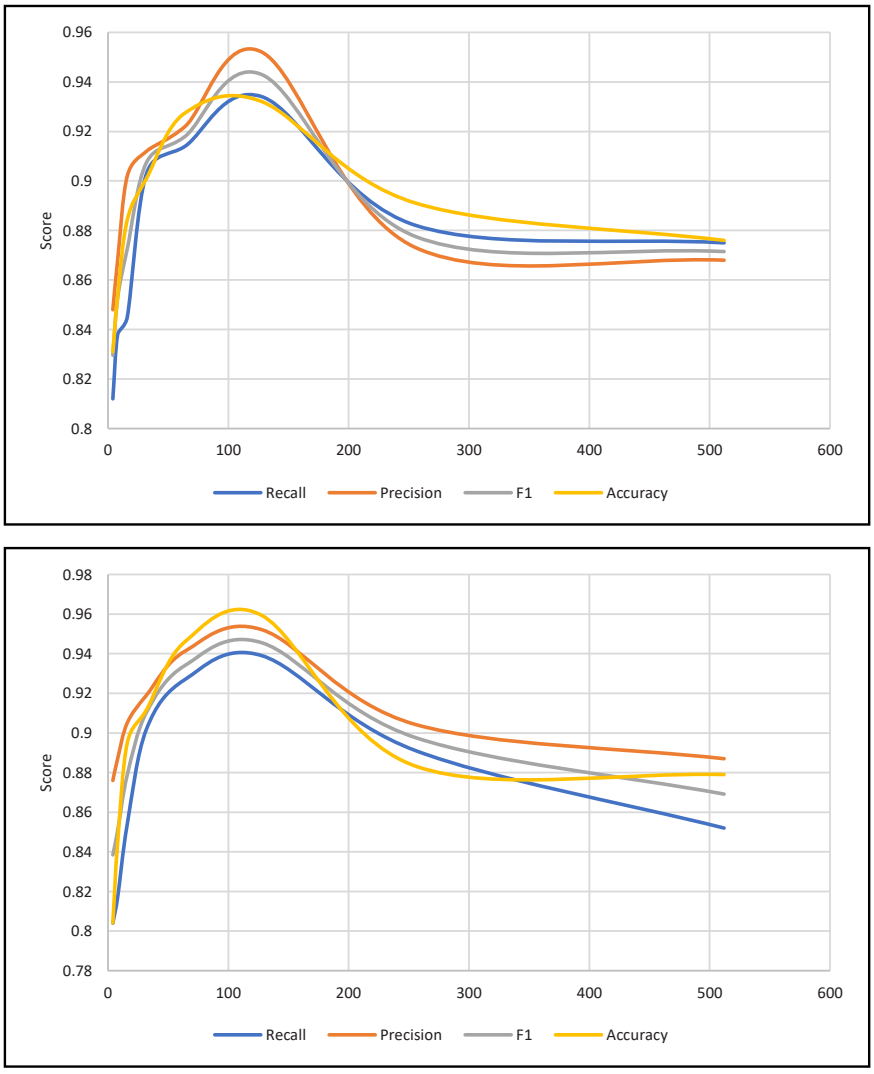

Fig. 9. (a). Impact of the sequence batch size on dataset 1. Fig. 9(b). Impact of the sequence batch size on dataset 2.

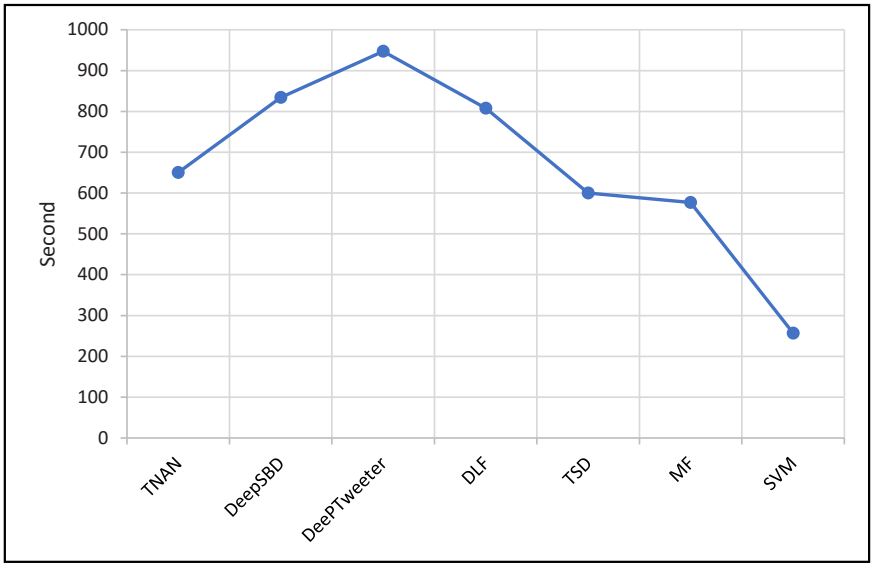

Fig. 10. Run time of the proposed model compared with those of the baseline methods.

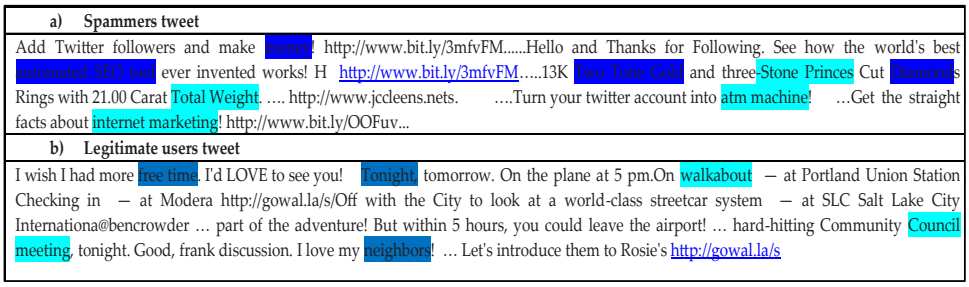

Fig. 11. (a and b). Example of attentive words in the Twitter content.

Funding

This work is funded by the Deputyship for Research and Innovation, Ministry of Education in Saudi Arabia through project number 445–9–793.

CRediT authorship contribution statement

Maged Nasser: Writing – review & editing, Writing – original draft, Methodology, Formal analysis, Data curation, Conceptualization. **Faisal Saeed:** Writing – review & editing, Supervision, Project administration, Methodology, Funding acquisition, Formal analysis, Conceptualization. **Aminu Da'u:** Writing – review & editing, Visualization, Validation, Software, Resources, Investigation, Data curation, Conceptualization. **Abdulaziz Alblwi:** Writing – review & editing, Validation, Project administration, Investigation, Funding acquisition, Formal analysis. **Mohammed Al-Sarem:** Writing – review & editing, Validation, Project administration, Investigation, Funding acquisition, Formal analysis.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors extend their appreciation to the Deputyship for Research and Innovation, Ministry of Education in Saudi Arabia for funding this work through the project number 445–9–793.

References

[1] R. Krithiga, E. Ilavarasan, A comprehensive survey of spam profile detection methods in online social networks, Proc. J. Phys.: Conf. Ser. (2019) 012111.
[2] X. Hu, J. Tang, Y. Zhang, H. Liu, Social spammer detection in microblogging, Proc. IJCAI Int. Jt. Conf. Artif. Intell. (2013) 2633–2639.
[3] T. Wu, S. Liu, J. Zhang, Y. Xiang, Twitter spam detection based on deep learning, Proc. ACM Int. Conf. Proc. Ser. (2017).
[4] S. Kaddoura, G. Chandrasekaran, D.E. Popescu, J.H. Duraisamy, A systematic literature review on spam content detection and classification, Peerj Comput. Sci. 8 (2022).
[5] V. Chinnaiah, C.C. Kiliroor, Heterogeneous Feature Analysis on Twitter Data Set for Identification of Spam Messages, Int. Arab J. Inf. Technol. 19 (2022) 38–44.
[6] Z. Alom, B. Carminati, E. Ferrari, A deep learning model for Twitter spam detection, Online Soc. Netw. Media 18 (2020).
[7] Ban, X.; Chen, C.; Liu, S.; Wang, Y.; Zhang, J. Deep-learnt features for Twitter spam detection. In Proceedings of the 2018 International Symposium on Security and Privacy in Social Networks and Big Data, SocialSec 2018, 2018; pp. 22-26.
[8] M. Fazil, A.K. Sah, M. Abulaish, DeepSBD: a deep neural network model with attention mechanism for socialbot detection, Ieee Trans. Inf. Forensics Secur. 16 (2021) 4211–4223.
[9] C. Johnson, B. Khadka, R.B. Basnet, T. Doleck, Towards detecting and classifying malicious urls using deep learning, J. Wirel. Mob. Netw., Ubiquitous Comput., Dependable Appl. 11 (2020) 31–48.
[10] A. Makkar, N. Kumar, An efficient deep learning-based scheme for web spam detection in IoT environment, Future Gener. Comput. Syst. - Int. J. Escience 108 (2020) 467–487.
[11] K. Archchitha, E.Y.A. Charles, Ieee, Opinion Spam Detection in Online Reviews Using Neural Networks, Sep 03-04. Proceedings of the 19th International Conference on Advances in ICT for Emerging Regions (ICTer), Univ Colombo Sch Comp, Colombo, SRI LANKA, 2019.
[12] P. Bhuvaneshwari, A.N. Rao, Y.H. Robinson, Spam review detection using self attention based CNN and bi-directional LSTM, Multimed. Tools Appl. 80 (2021) 18107–18124.
[13] Pennington, J.; Socher, R.; Manning, C.D. GloVe: Global vectors for word representation. In Proceedings of the EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 2014; pp. 1532-1543.
[14] Mikolov, T.; Yih, W.T.; Zweig, G. Linguistic Regularities in Continuous Space Word Representations. In Proceedings of the Proceedings of the 2nd Workshop on Computational Linguistics for Literature, CLfL 2013 at the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2013, 2013; pp. 746-751.
[15] Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K.; Assoc Computat, L. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the Conference of the North-American-Chapter of the Association-for-Computational-Linguistics - Human Language Technologies (NAACL-HLT), Minneapolis, MN, Jun 02-07, 2019; pp. 4171-4186.
[16] S. Ouni, F. Fkih, M.N. Omri, BERT- and CNN-based TOBEAT approach for unwelcome tweets detection, Soc. Netw. Anal. Min. 12 (2022).
[17] N. Peinelt, D. Nguyen, M. Liakata, tBERT: Topic models and BERT joining forces for semantic similarity detection. Proceedings of the Proceedings of the Annual Meeting of the, Association for Computational Linguistics, 2020, pp. 7047–7055.
[18] Chen, Q.X.; Yao, L.X.; Yang, J.; Ieee. SHORT TEXT CLASSIFICATION BASED ON LDA TOPIC MODEL. In Proceedings of the 5th International Conference on Audio, Language and Image Processing (ICALIP), Shanghai, PEOPLES R CHINA, Jul 11-12, 2016; pp. 749-753.
[19] Li, Y.X.; Iop. Short Text Classification Improved by Feature Space Extension. In Proceedings of the 5th International Conference on Electrical Engineering, Control and Robotics (EECR), Guangzhou, PEOPLES R CHINA, Jan 12-14, 2019.
[20] Y. Kontsewaya, E. Antonov, A. Artamonov, Evaluating the Effectiveness of Machine Learning Methods for Spam Detection, Proc. Procedia Comput. Sci. (2021) 479–486.
[21] R.M.K. Saeed, S. Rady, T.F. Gharib, An ensemble approach for spam detection in Arabic opinion texts, J. King Saud. Univ. -Comput. Inf. Sci. 34 (2022) 1407–1416.
[22] Ahmed, F.; Abulaish, M. An MCL-based approach for spam profile detection in online social networks. In Proceedings of the Proc. of the 11th IEEE Int. Conference on Trust, Security and Privacy in Computing and Communications, TrustCom-2012 - 11th IEEE Int. Conference on Ubiquitous Computing and Communications, IUCC-2012, 2012; pp. 602-608.
[23] Al-Zoubi, A.M.; Alqatawna, J.; Faris, H.; Ieee. Spam Profile Detection in Social Networks Based on Public Features. In Proceedings of the 8th International Conference on Information and Communication Systems (ICICS), Irbid, JORDAN, Apr 04-06, 2017; pp. 130-135.
[24] A.M. Al-Zoubi, H. Faris, J. Alqatawna, M.A. Hassonah, Evolving Support Vector Machines using Whale Optimization Algorithm for spam profiles detection on online social networks in different lingual contexts, Knowl. -Based Syst. 153 (2018) 91–104.
[25] C. Chen, J. Zhang, Y. Xie, Y. Xiang, W.L. Zhou, M.M. Hassan, A. AlElaiwi, M. Alrubaian, A Performance Evaluation of Machine Learning-Based Streaming Spam Tweets Detection, Ieee Trans. Comput. Soc. Syst. 2 (2015) 65–76.
[26] K.S. Adewole, T. Hang, W.Q. Wu, H.B. Songs, A.K. Sangaiah, Twitter spam account detection based on clustering and classification methods, J. Supercomput. 76 (2020) 4802–4837.
[27] J. Martinez-Romo, L. Araujo, Detecting malicious tweets in trending topics using a statistical analysis of language, Expert Syst. Appl. 40 (2013) 2992–3000.
[28] Al-Janabi, M.; De Quincey, E.; Andras, P. Using supervised machine learning algorithms to detect suspicious URLs in online social networks. In Proceedings of the Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2017, 2017; pp. 1104-1111.
[29] A. Da'u, N. Salim, Aspect extraction on user textual reviews using multi-channel convolutional neural network, Peerj Comput. Sci. (2019).
[30] G.C. Ruan, Y. Tan, A three-layer back-propagation neural network for spam detection using artificial immune concentration, Soft Comput. 14 (2010) 139–150.

[31] J. Ma, W. Gao, P. Mitra, S. Kwon, B.J. Jansen, K.F. Wong, M. Cha, Detecting rumors from microblogs with recurrent neural networks, Proc. IJCAI Int. Jt. Conf. Artif. Intell. (2016) 3818–3824.
[32] G.X. Xu, D.Q. Zhou, J. Liu, Social Network Spam Detection Based on ALBERT and Combination of Bi-LSTM with Self-Attention, Secur. Commun. Netw. 2021 (2021).
[33] A. Barushka, P. Hajek, Spam detection on social networks using cost-sensitive feature selection and ensemble-based regularized deep neural networks, Neural Comput. Appl. 32 (2020) 4239–4257.
[34] Gupta, H.; Jamal, M.S.; Madisetty, S.; Desarkar, M.S.; Ieee. A Framework for Real-Time Spam Detection in Twitter. In Proceedings of the 10th International Conference on Communication Systems and Networks (COMSNETS), Bangalore, INDIA, Jan 03-07, 2018; pp. 380-387.
[35] G. Jain, M. Sharma, B. Agarwal, Spam detection in social media using convolutional and long short term memory neural network, Ann. Math. Artif. Intell. 85 (2019) 21–44.
[36] S. Madisetty, M.S. Desarkar, A Neural Network-Based Ensemble Approach for Spam Detection in Twitter, Ieee Trans. Comput. Soc. Syst. 5 (2018) 973–984.
[37] P.K. Roy, J.P. Singh, S. Banerjee, Deep learning to filter SMS Spam, Future Gener. Comput. Syst. - Int. J. Escience 102 (2020) 524–533.
[38] Zhao, W.N.X.; Jiang, J.; Weng, J.S.; He, J.; Lim, E.P.; Yan, H.F.; Li, X.M. Comparing Twitter and Traditional Media Using Topic Models. In Proceedings of the 33rd European Conference on Information Retrieval, Dublin, IRELAND, Apr 18-21, 2011; pp. 338-+.
[39] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.
[40] Darling, W.M. A theoretical and practical implementation tutorial on topic modeling and gibbs sampling. In Proceedings of the Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies, 2011; pp. 642-647.
[41] Lu, J.S.; Yang, J.W.; Batra, D.; Parikh, D. Hierarchical Question-Image Co-Attention for Visual Question Answering. In Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS), Barcelona, SPAIN, 2016.
[42] Y. Guo, Z. Mustafaoglu, D. Koundal, Spam detection using bidirectional transformers and machine learning classifier algorithms, J. Comput. Cogn. Eng. 2 (2023) 5–9.
[43] W. Shishah, Fake News Detection Using BERT Model with Joint Learning, Arab. J. Sci. Eng. 46 (2021) 9115–9127.
[44] R.K. Kaliyar, A. Goswami, P. Narang, FakeBERT: Fake news detection in social media with a BERT-based deep learning approach, Multimed. Tools Appl. 80 (2021) 11765–11788.
[45] K. Lee, B. Eoff, J. Caverlee, Seven months with the devils: A long-term study of content polluters on twitter, Proc. Proc. Int. AAAI Conf. web Soc. Media (2011) 185–192.
[46] M. Slavin, Applications of Stochastic Gradient Descent to Nonnegative Matrix Factorization, University of Waterloo, 2019.
[47] Y. Zhu, X. Wang, E. Zhong, N. Liu, H. Li, Q. Yang, Discovering spammers in social networks, Proc. Proc. AAAI Conf. Artif. Intell. (2012) 171–177.