Journal of Ambient Intelligence and Humanized Computing (2021) 12:9287–9302
https://doi.org/10.1007/s12652-020-02640-5

ORIGINAL RESEARCH

TextSpamDetector: textual content based deep learning framework for social spam detection using conjoint attention mechanism

E. Elakkiya[1] · S. Selvakumar[1,2] · R. Leela Velusamy[1]

Received: 8 July 2020 / Accepted: 24 October 2020 / Published online: 9 November 2020

Abstract
Online Social Networks (OSNs) allow easy membership leading to registration of a huge population and generation of voluminous information. These characteristics attract spammers to spread spam which may cause annoyance, financial loss, or personal information loss to the user and also weaken the reputation of social network sites. Most of the spam detection methods are based on user and content-based features using machine learning techniques. But, these annotated features are difficult to extract in real-time due to the privacy policy of most social network sites. Even for the features that can be extracted, because of their large size, the manual extraction process is complex and time-consuming. So there is a need for text level spam detection that does not require extraction of hard-core features. Existing deep learning based or existing single attention mechanism based text classification methods could not perform well as social network data are sparse with short texts and noises. Moreover, Spammers avoid direct spam words and use indirect words to evade spam filtering techniques and thus resulting in the dynamic and non-stationary nature of the social network spam texts. These indirect words contain hidden context that creates attention drift problem. So conjoint attention mechanism along with two attention mechanisms namely normal attention and context preserving attention are proposed to avoid attention drift problem in this deep learning-based text level spam detection technique (TextSpamDetector). Attention drift problem is solved by one attention mechanism which helps to find the important words while another attention mechanism allows focusing on attention in target context by referring to higher level abstraction of context vector. These attention mechanisms are referring to different context representations of the input text for finding informative words from the structural context representation. This structural context representation containing both local semantic features as well as global semantic dependency features is generated by CNN and BiLSTM. The proposed model is evaluated with the existing spam detection techniques using three datasets and the experimental results have proved that the proposed model performs well in terms of accuracy, F measure, and false-positive rate.

Keywords Social network spam detection · Attention network · Text classification · Deep learning

1 Introduction

OSNs are very popular nowadays and contain a large population as it allows people to communicate with each other almost instantaneously. The people using OSN sites now are 3.8 billion which is one-third of the entire world population (Simon 2020). Some OSNs allow known people to communicate and some OSNs allow even strangers to communicate with each other based on their interests, location, and thoughts. These characteristics attract cybercriminals for their malicious activities in OSN including the spreading of spams. Spam contains unwanted information, malicious URL links that direct to malicious sites for downloading malware or stealing personal information for identity theft. Spam not only disturbs individuals but also diminishes the reputation of OSNs. So the researchers and some of the OSNs, such as twitter itself propose spam detection methods to protect the OSN users.

Conventional spam detection methods focus on blacklisting methods and applying machine learning with the user and content-based features. These feature-based spam

✉ E. Elakkiya
406115004@nitt.edu

[1] Computer Science and Engineering, National Institute of Technology, Tiruchirappalli, India

[2] Indian Institute of Information Technology, Una, Himachal Pradesh, India

detection methods are based on user behaviour attributes such as the total number of tweets posted per day, the total number of followers and content-based attributes such as the number of hashtags in a tweet, number of people mentioned in the tweet. These features can be easily manipulated by the spammers and can evade from the spam detection approaches.

Also, these machine learning methods perform well on email spam detection methods as email has clear long text but the detection rate is less in OSNs due to the nature of short texts and presence of noisy texts. Most of the machine learning algorithms consider the user and content-based features for spam detection (Zhang et al. 2016; Egele et al. 2015; Chen et al. 2016) but it is difficult to extract the values for the features in real-time. So, a more efficient text-based spam detection method is required instead of a traditional feature-based spam detection technique to find the hidden relationship in social network data as the OSN platform has high data volatility.

Existing text-based classification methods use conventional methods such as TFIDF, count vectorizer, etc., for finding the keywords from the text which is helpful to determine the category of the text. But these methods do not preserve the sequential structure of the text. So deep learning methods such as LSTM are helpful to retain the sequential structure. But not all the words in the text are important to find the spam. So a method is required to identify the important sensory words from the spam text. This can be accomplished by the attention method as it is based on the assumption that human recognizes an object by focusing on only selected parts of the whole perception space. Attention captures the significant features in a sentence by learning their weights automatically. The weight of each word measures the extent that the word is influencing the meaning of a sentence. But the existing attention methods such as (Bahdanau et al. 2014; Luong et al. 2015; Vaswani et al. 2017) suffer the attention drift problem (Cheng et al. 2017) for data that contains the non-stationarity and hidden context. Non-stationarity indicates that the data changes over time and hidden context means the words are not given explicitly in the form of predictive features. Non-stationarity and hidden context have become mostly prevalent in the spam text as spammers do not use explicit spam words and change the way of writing spam to look legitimate to evade from the spam filtering techniques. So, an improved attention mechanism is required to solve the attention drift problem and extract important semantic information from the hidden context environment. Hence this proposed framework for text-level spam detection contains three layers. In the first layer, CNN extracts the local context features and n-gram information of the sentence. CNN does not capture the sequential order of the text. But LSTM remembers the long sequences of text that are mainly required for spam text classification. LSTM has been shown to yield good results in text classification (Rao et al. 2018), but it considers only the forward direction and exploits the preceding context alone. So in the second layer, BiLSTM which is an extension of LSTM is used to train the forward hidden layer and also the backward hidden layer that can produce both historical and future context dependencies.

Different level context representations are created by CNN and BiLSTM in these two layers for producing comprehensive structural context representation containing both local semantic features as well as global dependent information features. But these different level context representations may lead to the loss of data or flatten the information as the social spam texts are short in length. Even if the loss is small it may lead to text misjudgement. But CNN is required before BiLSTM for decreasing the dimensionality of sentence otherwise BiLSTM will increase the number of network parameters that increase the network complexity and difficulty to optimize. Therefore, the need for a mechanism that focusses the important words in the context and extracts without any information loss, arises and is justified. So conjoint attention mechanism is proposed in the third layer to avoid the information loss and solve the attention drift problem caused by hidden context in the spam text. This is accomplished by conjoint attention which uses two attention namely, normal attention and context preserving attention mechanisms referring to different level context vectors that are applied together. Attention drift problem is solved by one attention mechanism that helps to find the important words while another attention mechanism allows focusing on attention in target context by referring to different higher level abstract version of the context vector. The justification for using these two attention mechanisms is that some studies have shown (Conneau et al. 2017) that multiple views of the same input sentences make the model learn that part of the sentence which is more important for the given task.

The sentence representation generated by BiLSTM is given as input to the normal attention mechanism to find the global sentence semantics. The important part of this is that it not only depends on the last state but depends on the weighted combination of all the states to avoid the long-range dependency problem and information loss. Context preserving attention mechanism is employed with the higher-level abstract version of the input sentence generated by CNN and the reference for the sentence representation is generated by the BiLSTM. Context preserving attention provides the local context and long-range dependent context information thus improving the learning capability of the model. Overall in this work, the model decides on what to attend to, based on different versions of the input sentence and what it has produced so far. Hence, this model extracts more information contained in the spam text to extract

semantic information from the hidden context for creating a more representative feature vector for the sentence. This text-based spam detection framework applies to all OSNs such as Facebook, Twitter, YouTube comments, Sina Weibo, etc.

The main contributions of this work include the following propositions:

1. Text-based spam detection which does not require historical information of a user.
2. The first framework that uses an attention network in spam detection and brings out attention drift problem in social spam text.
3. Conjoint attention for solving attention drift problems, avoiding information loss, and bringing semantic information from the hidden context.
4. Two kinds of attention mechanisms referring different levels of context vectors to preserve comprehensive information such as local context, preceding, and succeeding context relation.
5. A new framework for CNN and BiLSTM based deep learning approach for spam detection in text level.
6. User and content feature independent model for spam detection of privacy-preserving social network sites without domain expertise and hardcore feature extraction.

The rest of this paper is structured as follows: Sect. 2 discusses the related works. Section 3 gives the details about the proposed approach. The experimental evaluation of the method is discussed in Sect. 4. Section 5 concludes the paper by summarizing the contributions.

2 Related work

The existing spam detection methodology can be classified into two categories such as spammer detection, and individual spam detection. In the first category, most of the researchers focused on URL blacklist methods and machine learning methods. URL links present in the tweet are used by the spammers to perform their malicious activities. One of the cost-effective methods for spam detection is the URL blacklisting method. Spam detection is done by comparing the content in the tweet with the URL of the tweet (Benevenuto et al. 2010). It is considered as spam if they are not related. SVM classifier is used for the final classification. But it is better to analyze other powerful machine learning algorithms for good performance.

Monarch (Thomas et al. 2011) is a URL spam filtering real-time system to detect spammers based on URL analysis. URL related features were collected from the web browser, IP analysis, and DNS solver and the monarch does not rely on any machine learning classifiers. Analysis of how spam URL links were posted at the sender side and received at the receiver side was done (Cao and Caverlee 2014) and it was observed that spammers find difficulty in manipulating posting and click-based patterns rather than content and network features. But URL shortening service such as bit.ly alone was considered and other URL shortening services were not considered for analysis. URL and social network features were used to find malicious links in social networks (Alghamdi et al. 2016). Lexical, domain, and hostname features related to URLs with user and post based features were employed with different machine learning algorithms to identify the best model. However, this system is vulnerable to manipulation done by the spammers. Even though the URL blacklist method is simple, studies show that users click the link before it is added to the blacklist.

Machine learning methods in spammer detection were based on user behavior and the content-based features (Liu et al. 2017; Feng et al. 2018; Rathore et al. 2018; Ala'M et al. 2018; Jose and Babu 2019). Statistical analysis on 14 generic features was derived by the authors in (Ahmed and Abulaish 2013) on three categories, viz., interaction, URL, and post related features applicable to both Facebook and Twitter. These features were applied to the basic classification algorithms for further detection. Additionally, Markov clustering was applied to weighted graphs to identify the spam campaign. The drawback of this method was the spam campaign of the cluster containing three to eight profiles of the same user. Evasion tactics performed by the spammer to avoid spam detection techniques are analyzed by (Yang et al. 2013) in which new features were added with the existing features and categorized as graph-based, automation based, neighborhood-based, and timing-based features. But the correlation between the features is not considered. Another user and content-based features used spammer detection method (Zheng et al. 2015) and found two more additional new features such as the number of days from creation and an average number of comments. These features were applied such as SVM, Decision Tree, Naïve Bayes, and Bayes Network machine learning algorithms, out of which the SVM classification algorithm performed well. But almost all the spam detection methods extract the features manually which may be difficult for real-time extraction.

A heterogeneous information network containing products, reviews, users, and spam features was employed in (Shehnepoor et al. 2017) for the online review of the spam dataset. This method considered user behavior and linguistic-based features and analyzed the supervised and unsupervised methods. Weight for each feature was calculated and the probability of estimation to be the spam was calculated using the machine learning algorithms. Word, content, and user-based features were used to detect spam in YouTube comments by incremental learning and topic

modeling in (Song et al. 2017). Topic modeling was used to extract the semantic meaning in user comments. But in this work video related attributes were not considered.

Another category of spam detection is content-based spam detection. Statistical analysis of words in trending topics applied to tweets to identify the spam without considering the user based features was discussed in (Martinez-Romo and Araujo 2013). HSpam14 dataset containing 14 million tweets were created by (Sedhai and Sun 2015). A semi-supervised method for spam detection (Sedhai and Sun 2017) was used by employing four lightweight detectors to filter the spam at the text level. Another tweet based spam detection method was introduced in (Chen et al. 2015) and the performance of different machine learning classifiers was examined. But all these methods used the features derived from the content instead of applying the raw text. Some of the deep learning methods (Wu et al. 2017a, b) were also proposed for spam detection and found that the results are better than the traditional machine learning classifiers like SVM, Naïve Bayes, etc. CNN and LSTM based semantically pre-trained tweets were used to classify a message as spam or legitimate in (Jain et al. 2019). Ensemble-based spam detection considering both user and content-based features were used in (Madisetty and Desarkar 2018). They combined the feature-based method using the best of SVM and Random forest classifier and CNN based text level spam detection. The drawbacks of most of the methods were the time consumed to extract the features and difficulty in extracting it in real-time as almost all social network sites are privacy preserving websites. In our proposed framework, word embedding features are used as the universal features which do not require any historical information and manual intervention.

3 Proposed methodology

The problem identified in this paper is as follows: Given a social media text T, classify whether it is a spam or not. The entire learning algorithm is depicted in Fig. 1 and contains five modules such as embedding module, feature extraction module, semantic dependency extraction module, conjoint attention mechanism module, and classification module. Embedding module is used to encode input text into a numerical vector representation. The feature extraction module is used to extract the local information held in every position of the sentence. In the semantic dependency extraction module, BiLSTM neural network is used to extract the whole sentence semantics by finding the preceding and succeeding context dependency. The conjoint attention mechanism works as follows: Normal attention is first applied to the sentence representation returned by the semantic dependency extraction module. Then context preserving attention is calculated by finding the similarity between the local representation and semantic dependent context representation for finding comprehensive information viz., local context, historical context and future context of the sentence. Local representation is generated by CNN with the max-pooling operation and semantic dependent context representation is generated by BiLSTM. The features generated by the conjoint attention mechanism are fed into the final classification module and softmax function is applied for classifying spam and non-spam text instances. The overall architecture of the proposed model is depicted in Fig. 2.

3.1 Word embedding

The dataset D is constructed with n number of social media texts (for example Tweet or YouTube comment) which comprises of spam text and normal text. The

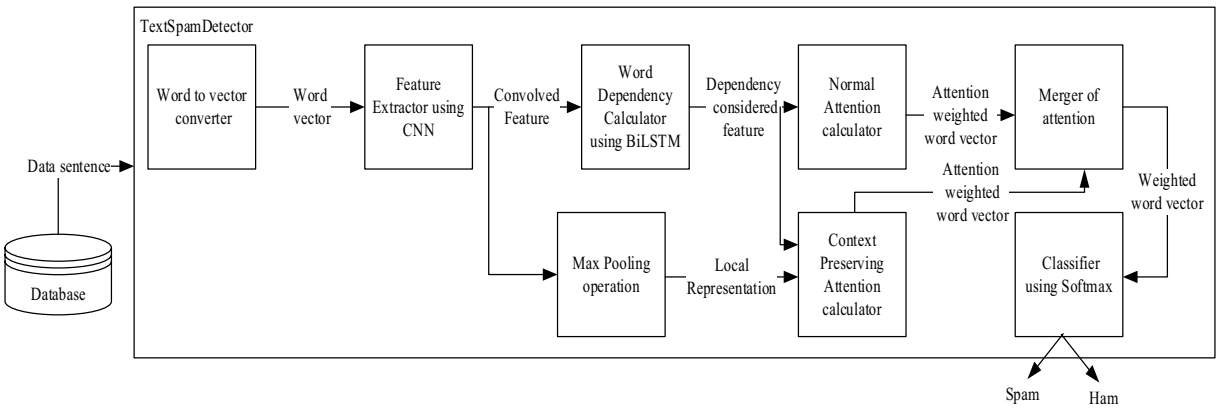

Fig. 1 Block Diagram of the TextSpamDetector

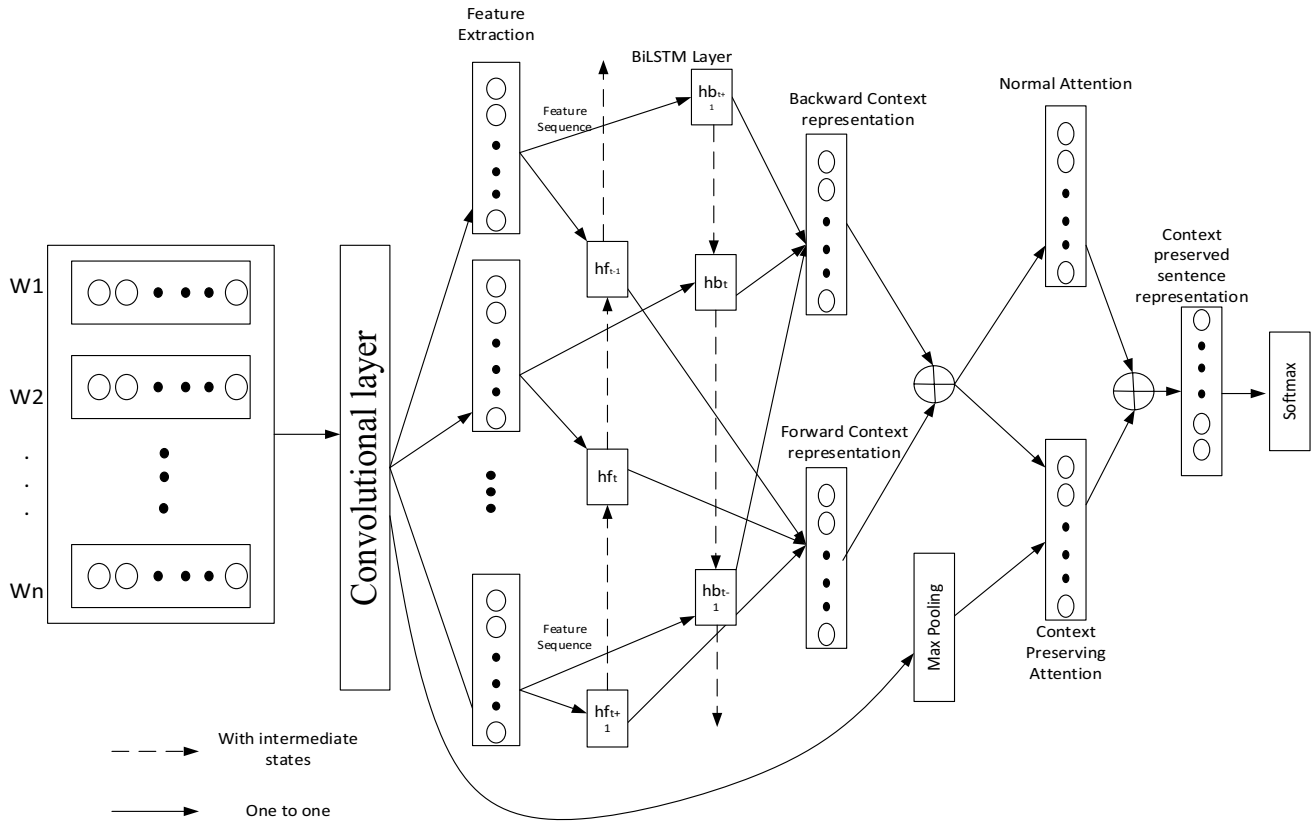

Fig. 2 Architecture of the TextSpamDetector

features used to classify spam text are the words contained in it. For further classifying the text, it should be represented in the numerical form. There are two methods used for vector representation viz., one-hot encoding and dense vector representation. In one-hot encoding, every word is associated with one integer index. If the word is present in the text then the integer index value into a binary vector length of N (size of the vocabulary) is assigned with 1 otherwise assigned with 0. For example, if the vocabulary is {this, I, like, most, pay, new, phone} then the word "phone" will create a binary vector as {0, 0, 0, 0, 0, 0, 1}. But it requires high dimensions and there is no semantic relationship between words. Similar words are treated as different in one-hot encoding. In dense representation, similar words can have similar vector representation and will also result in lower dimensional space so computational cost is less. So, dense representation is used in this work for the word to vector conversion. The popular and powerful representation word2vec is used in this paper and the detailed algorithm is described in (Mikolov et al. 2013a, b). Word2vec is trained with the skip-gram model by maximizing the average log probability of all the words. It predicts the range concerning the surrounding words of the current word whose input comes from the current word.

The social text vector is created with the words from the social text. The word to vector matrix is s × n, where s means the dimension of word vector and n is the length of the social media text. This social text matrix is fed into the convolutional layer for further processing. All punctuation and special characters are treated as separate word tokens and no preprocessing is done on social text since they are normally short in length and contain some other format of text structures like mention, hashtags, URL, etc.

3.2 Feature extraction module

CNN is used to extract robust and abstract features from the words (Banerjee et al. 2019). CNN consists of a sequence of convolution and pooling operations. Different sizes of kernels (i.e., 2-g, 3-g, 4-g, and 5-g) are used for convolution to obtain sufficient important features. A convolution operation is applied over the social text matrix $ST \in Real^{d \times |m|}$ and kernel $K \in Real^{d \times |s|}$ where d is the dimension, m is the number of words present in the

social text, and s is the size of the kernel. Feature vector with the dimension of (|m|-K + 1) is computed as follows:

$$ST = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1d} \\ w_{21} & w_{22} & \cdots & w_{2d} \\ \vdots & \vdots & \cdots & \vdots \\ w_{m1} & w_{m2} & \cdots & w_{md} \end{bmatrix}$$

$$\odot K = \begin{bmatrix} k_{11} & k_{21} \\ k_{12} & k_{22} \\ \vdots & \vdots \\ k_{1d} & k_{2d} \end{bmatrix}$$

where Θ is the convolution operator, *ST* is the social text matrix consisting of *m* number of embedding vectors and each vector is derived from each word. The social texts (e.g., tweet) are of different lengths and CNN does not accept inputs having different lengths. So, post padding techniques are used to create the messages in equal length. *K* is the kernel matrix and the output feature matrix is calculated using the convolution operation (Sarıgül et al. 2019) as follows:

$$\begin{aligned} f_1 &= w_{11}k_{11} + w_{12}k_{12} + \cdots + w_{1d}k_{1d} \\ &\quad + w_{21}k_{21} + w_{22}k_{22} + \cdots + w_{2d}k_{1d} \\ f_2 &= w_{21}k_{11} + w_{22}k_{22} + \cdots + w_{2d}k_{1d} \\ &\quad + w_{31}k_{21} + w_{32}k_{22} + \cdots + w_{3d}k_{1d} \\ f_m &= w_{(m-1)1}k_{11} + w_{(m-2)2}k_{12} + \cdots + w_{m1}k_{21} \\ &\quad + w_{m2}k_{22} + \cdots + w_{md}k_{2d} \end{aligned} \tag{1}$$

The feature matrix F consisting f_1, f_2, …, f_m in the dimension $(m - 2 + 1) \times 1$ is passed through the activation function. The resultant matrix F' in the dimension $(m - 2 + 1) \times 1$ represents the hidden features extracted from the social text. Furthermore, to derive only the important features from the hidden features, pooling operation is performed. This operation both shortens the training time by removing the low activation information and combats the overfitting which is created from the noisy text. In this work, max-pooling is used with the window size w. The max-pooling operation slides a window over its input and finds the maximum value in the window. For example, if the value of w = 4, then the maximum value is drawn among the 4 features.

The number of important features extracted from the feature matrix F is calculated in Eq. (2) as follows:

$$N = \frac{\text{Number of } f_i}{w} \quad i = 1 \text{ to } m \tag{2}$$

The feature vector $F` = [f_1', f_2', \ldots f_m']$ is given as input to the context preserving attention mechanism and feature vector $F = [f_1, f_2, \ldots, f_m]$ is given as input into the BiLSTM neural network.

3.3 Semantic dependency extraction module

CNN provides the abstract deep features without noise that are extracted from the convolution before pooling operation is passed to the next BiLSTM, which is a variation of LSTM. CNN is unable to capture the long dependency of the text (Zhou et al. 2016), which is mainly required for spam text classification. LSTM neural network is a type of recurrent neural network (RNN) containing a directional loop that can memorize and remember the past information. LSTM uses a gating mechanism to avoid vanishing gradient problem of traditional RNN.

The LSTM unit has three logic gates, viz., input gate, forget gate, and output gate. The input gate is used to decide what information needs to pass through the memory cell, forget gate is used to choose what information should be removed, and the output gate calculates the final information that is passed to the next state. The first step is to calculate what and how much information needs to be discarded from the memory using the forget gate and is computed as in Eq. (3).

$$fg_t = \sigma\left(W_{fg}\left[h_{t-1}, x_t\right] + b_{fg}\right) \tag{3}$$

The next step is to determine what information is allowed to pass through the memory cell. This contains two parts and is calculated using Eqs. (4) and (5). The first part calculates what information will be retained in the cell state using a sigmoid function. Sigmoid layer produces the output between 0 and 1 and the output 0 means "let no information pass" and 1 means "let all information pass". The second part uses the tanh layer to calculate the new state using h_{t-1} and x_t. These two parts will be passed to the third step having an update gate.

$$ig_t = \sigma\left(W_{ig}\left[h_{t-1}, x_t\right] + b_{ig}\right) \tag{4}$$

$$\widetilde{N}_t = tanh\left(W_N\left[h_{t-1}, x_t\right] + b_N\right) \tag{5}$$

The third step is to update the old cell state N_{t-1} into a new cell state N_t using the information already calculated in the previous state as depicted in Eq. (6).

$$N_t = fg_t \times N_{t-1} + ig_t \times \widetilde{N}_t \tag{6}$$

The final step is to decide the information going to output using Eqs. (7) and (8). This output gate also contains the sigmoid layer part and tanh layer part to enhance the nonlinearity of the network. Sigmoid layer chooses what information is going to output and this value is multiplied with tanh which squeezes the values between 1 to − 1 of the new cell state.

$$og_t = \sigma\left(W_{og}\left[h_{t-1}, x_t\right] + b_{og}\right) \tag{7}$$

$$h_t = og_t \times tanh\left(N_t\right) \tag{8}$$

where W_{fg},W_{ig}, W_N, and W_{og} denote the weight matrices, b_{fg}, b_{ig}, b_N, and b_{og} are the bias vectors, and σ (sigmoid), tanh (hyperbolic tangent) are the activation functions and are computed as in Eq. (9) and (10).

$$\sigma(\alpha) = \frac{1}{1 + e^{-\alpha}} \tag{9}$$

$$\tanh(\alpha) = \frac{e^{\alpha} - e^{-\alpha}}{e^{\alpha} + e^{-\alpha}} \tag{10}$$

The sequences of sentences are given as input to the LSTM unit along with previous LSTM unit output. This is repeated for each sentence and LSTM computes the important features of sentences in this way.

The conventional LSTM neural network makes use of preceding context relation obtained from forwarding parts of a sentence. But succeeding context relation is also required for the complete understanding of the spam detection problem. As a result, BiLSTM uses both the past words context relation and future words context relation by composing two independent LSTM, each of which combines the information from the forward and backward direction of a sentence. At time t, the forward LSTM calculates the hidden vector, hf_t calculated from the previously hidden vector, hf_{t-1} and the input vector, x_t and backward LSTM hidden vector, hb_t calculates the previously hidden vector, hb_{t-1} and the input vector, x_t. The final hidden vector, h_t is the combined vector of both forward and backward hidden vectors as depicted in Fig. 2.

3.4 Attention

All the contextual words in a sentence are not equally important for spam classification. Some spam words play a decisive role in spam detection. Attention allows the network to refer back to the input sentence. Technically, attention is quantifying the interdependence between the input and output elements. In this paper, Attention is used to map each word in the output Social Text (ST) to the important and relevant words from the input ST and assign higher weights to the important words in the output of ST thus improving the accuracy of the model.

Spammers are normally avoiding the direct spam words in the spam text to skirt from the spam filtering techniques. So there is a need for extracting implicit spam words from the text that do not contain any explicit spam words. Single attention model cannot accurately associate each feature vector with the corresponding target vector in the input text and finds difficulty in capturing sensory words. This is called attention drift. This motivated us to develop some mechanism to focus on the right important words in the target context. So conjoint attention mechanism is proposed to solve the attention drift problem where one attention mechanism helps to find the important words while another mechanism allows focusing on attention in target context by referring to different context vectors. For achieving this, as discussed in the previous section, hidden feature vector representation extracted from the CNN using the input context vector before applying pooling operation is fed into the BiLSTM and the BiLSTM produces the global semantic dependency feature representation as ĉ. After that, ĉ is compared with the local semantic feature representation generated by CNN with max pooling operation for extracting the most important words that are used to detect the spam effectively. The attention mechanism applied between the local semantic feature and global semantic dependency feature is called context preserving attention. The attention mechanism applied to the global semantic dependency feature representation and input text context vector is named as normal attention in this work.

3.4.1 Normal attention

Normal attention maps the output ST of the BiLSTM neural network to the important words of input ST and assigns higher weights to the words in output ST that have a higher influence on the semantics of the text. The important part of this is that it not only depends on the last state but also depends on the weighted combination of all the input states. The reason behind this is that RNN is a biased model, where latter words are more dominant than earlier words. Therefore, to avoid this, BiLSTM is used but it also may lead to loss of some information that is really required to represent the actual semantics of the text due to the presence of indirect spam words in ST. If BiLSTM makes a bad summary of the social text, then the prediction accuracy becomes less. Since it is observed that LSTM/RNN has a "long-range dependency problem" when it tries to encode longer sentences it tends to become forgetful in specific cases even though BiLSTM captures long-range dependency better than RNN. So the proposed normal attention mechanism employs previous state information also with the final output and it refers back to the important input text.

The normal attention is calculated as follows: The past and future context representation of all the states $\hat{c}_i$ where i = 1 to number of LSTM states, generated by BiLSTM is fed into the single perceptron to compute the hidden representation $\vec{h}$ as follows in Eq. (11):

$$\vec{h} = \tanh(w\hat{c}_i + b) \tag{11}$$

where w is the weight, b is the bias in the neuron, tanh is the hyperbolic tangent function. Then the similarity between the $\vec{h}$ and word-level context vector $\vec{v}$ is computed using softmax function and formulated as follows in Eq. (12) and Eq. (13):

$$ws = \frac{\exp(b_i)}{\sum_{i=1}^{n} \exp(b_i)} \tag{12}$$

$$b_i = \text{CosineSim}\left(\vec{h}, \vec{v}\right) \tag{13}$$

where n is the number of words in the text and $\vec{v}$ is the word level context vector contains the high-level representation of the important words. It is initialized randomly and learned together with the training process. The output of normal attention is computed using Eq. (14) as follows:

$$o_s = \sum_{i=1}^{n} \hat{c}_i ws_i \tag{14}$$

Normal attention assigns more weights to the words in the output that have high similarity with the word-level context vector. Moreover, the word-level context vector is produced using random initialization and learned jointly. This implies that the word-level context vector is not generalizable and independent of input text which requires different level representations of input sentences to correctly focus on the important words, justifying the need for the following attention mechanism.

3.4.2 Context preserving attention

Some studies have shown (Conneau et al. 2017) that multiple views of the same input sentences make the model learn that part of the sentence which is more important for the given task. So context preserving attention is used to give a more detailed analysis of important words by referring to another context representation. Instead of referring to traditional input text, this Attention mechanism refers to the output of text representation generated by CNN with max-pooling, thus capturing the hierarchical abstractions of an input sentence. So context preserving attention maps the output of BiLSTM with the higher-level abstract version of input sentence generated by CNN with max-pooling and helps the model with a better understanding of the important words. These important words contain the characteristics of local context representation, succeeding context relation, and preceding context relation since these representations are retrieved from CNN and BiLSTM. In this proposed context preserving attention mechanism, the final state of BiLSTM is only used as the output representation to reduce the redundancy and the computation time as the previous normal attention mechanism used all the weighted combination of states.

Algorithm 1 Pseudocode of TextSpamDetector
Input: N variable-length sentences in dataset D, and their labels
Output: Classified label
1. Construct the sentence matrix using each sentence in D using word2vec [30-31] as explained in section 3.1.
2. For i in [1: m] do
   a. Convolution layer is used to obtain the local representation L for i^{th} sentence in the feature extraction module
   b. Bidirectional long short term memory is used to obtain the past and future context representation ĉ from L
   c. Max pooling operation is employed in L
   d. The normal-attention mechanism is applied to ĉ and obtain the informative word context o_s using equations (11-14)
   e. Context preserving attention weights are calculated from L and ĉ and obtain the informative word context o_h using equations(15-18)
   f. Obtain the final informative sentence representation by concatenating o_h and o_s s = [o_s , o_h]
   g. End for
3. The final feature vector s is applied to the softmax layer to obtain the class label

The context preserving attention mechanism is calculated as follows: attention weights are computed by comparing the local context representation L generated by the CNN with the ĉ generated by the BiLSTM and higher similar words having higher weights and lower similar weights having lower weights. The local context representation generated by CNN after max pooling is represented in Eq. (15) as follows:

$$L = \left[L_1, L_2, \ldots, L_n\right] \in Real^{k \times |n|} \tag{15}$$

Cosine similarity is used as the similarity function for calculating attention weights as given in Eq. (16):

$$a_i = \text{CosineSim}\left(L_i, \hat{c}\right) \tag{16}$$

Attention weights are calculated as in Eq. (17):

$$wh_i = \frac{\exp\left(a_i\right)}{\sum_{i=1}^{k} \exp\left(a_i\right)} \tag{17}$$

The output of this context preserving attention is given in Eq. (18):

$$o_h = \sum_{i=1}^{n} \hat{c}_i wh_i \in Real^k \tag{18}$$

The final context representation that contains the most informative words $s = [o_s, o_h]$ is obtained by merging the two feature vectors of attention mechanisms.

3.5 Output layer

The main drawback of the neural network is the overfitting with the low volume of data and the solution for this problem is reducing the size of the network instead of adding more data. The dropout layer is a regularization technique used to avoid overfitting as this layer randomly drops units. The final informative words comprising the context sentence representation are fed into the dropout layer and the softmax layer is added to give the spam class. In this work, cross-entropy is used as a loss function and it is normalized with L2 regularizer and applied to Θ to avoid overfitting and computed as in Eq. (19).

$$\text{loss} = -\sum_{j}^{m}\sum_{k}^{c} y_{jk} \log\left(\hat{y}_{jk}\right) + \lambda\|\theta\|^2 \tag{19}$$

where m is the number of spam sentences, c is the number of classes, y is the actual label, and $\hat{y}$ is the predicted label. Adam optimizer is used as the training algorithm. The overall learning algorithm of the proposed TextSpamDetector is summarized in Algorithm 1.

4 Experimental settings and results

The proposed approach was implemented on the Keras 2.0 API using Python 2.7 with a Tensor flow backend on 4 GB RAM under Windows 10.

4.1 Datasets

Three datasets have been used for the evaluation of the proposed approach. A Twitter dataset which is available in ("UtkMl's Twitter" 2019) has been used. Some text-based datasets for Twitter are also available but it contains the tweet id and labels only. The text of tweet ids should be retrieved from the public stream API available on Twitter. But this is no longer useful as Twitter will discard the content after some period and the server returns nothing. This Twitter dataset used in our approach contains eight columns but the attributes such as tweets and labels are only used in our approach because this model does not use any spammer oriented features. Moreover to examine the efficiency and applicability of the proposed approach, SMS spam dataset available in UCI machine learning repository which replicates the short text of tweets in Twitter has been used for evaluation. This dataset was collected from various resources such as Grumbletext, UK public Forum and presented in (Tagg 2009). Another standard spam dataset based on YouTube social network sites (Almeida et al. 2016) which is available in the UCI data repository has been used for performance comparison. The statistical information of the dataset is given in Table 1.

4.2 Performance evaluation metrics

The performance evaluation of the proposed approach is based on the standard classification metrics such as Accuracy, F measure, and false positive rate (FPR) calculated using the following Eqs. (20–24). These metrics calculated from the confusion matrix make use of measures such as True positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). TP is the number of spam instances that are correctly classified as spam, FP is the number of non-spam instances that are incorrectly classified as spam, TN is the number of non-spam instances that are correctly classified as non-spam, and FN is the number of spam instances that are incorrectly classified as spam.

The false-positive rate is the measure of the fraction of non-spam instances that were incorrectly classified as spam. Accuracy is the fraction of the spam instances that were found correctly among all the instances. F measure is the harmonic mean of precision (P) and recall (R). Precision is the fraction of data instances predicted as positive that is actually positive. Recall measures the capability of the model to predict the spam.

$$FPR = \frac{FP}{FP + TN} \tag{20}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{21}$$

Table 1 Dataset summary statistic

Dataset	Total instances	Training instances		Testing instances	
		No. of non-spam	No. of spam	No. of non-spam	Noof spam
Twitter	11,968	4102	3876	2052	1938
YouTube	1956	634	670	317	335
SMS	5574	3218	498	1609	249

$$FMeasure = 2 \times \frac{P \times R}{P + R} \tag{22}$$

$$P = \frac{TP}{TP + FP} \tag{23}$$

$$R = \frac{TP}{TP + FN} \tag{24}$$

4.3 Experimental results

The proposed approach has been compared with the baseline methods used for text classification along with some state of the art neural network methods, existing spam detection methods, and existing attention mechanisms. The parameters of the existing methods are tabulated in Table 2 and for the remaining methods, the default parameters have been assigned.

The existing methods used for comparison of the proposed method, TextSpamDetector are as follows:

4.4 Baseline methods

SVM—support vector machine classifier uses a bag of bigrams as features.

KNN—K nearest neighbor classifier uses a bag of bigrams as features.

Naïve Bayes—Naïve Bayes classifier uses a bag of bigrams as features.

Random Forest—Random forest is also a baseline method that uses the bag of words feature.

Random Tree—Random Tree is also a traditional baseline method that uses the bag of words feature.

4.5 Neural network methods

CNN (Kim 2014)—Convolved feature with max-pooling operation using CNN is fed into the fully connected network and is used for classification.

LSTM (Rao 2018)—Long short term memory is a type of RNN that uses a gating mechanism. The input text is fed into LSTM and the output of the last hidden state is the feature vector for the final classification layer.

BiLSTM (Xu et al. 2019a)—It consists of two parallel LSTMs, one of the input sequences and another one in the opposite direction and the combined last hidden state feature vector is used for the final classification.

C-LSTM (Zhou et al. 2015)—A combined model of CNN and LSTM is used for sentence representation and text classification.

DECNN (Xu et al. 2019b)—CNN with attention mechanism is used for text classification.

DBB-RDNN-Rel (Barushka and Hajek 2018)—Multilayer perceptron neural network with rectified linear units using tf-idf feature selection for spam detection.

SSCL (Jain et al. 2019)—A combined model of CNN and LSTM for sentence representation and text classification for spam detection.

4.6 Attention mechanism

In order to evaluate the impact of conjoint attention mechanism on the performance of TextSpamDetector, the experiments are conducted with some existing attention mechanisms. The base model of the first two layers are retained and conjoint attention mechanism in the third layer is replaced with existing attention mechanism such as simple attention, self-attention, and global attention for performance comparison. The context attention mechanism is applied as given in their literature (Feng et al. 2019).

Table 2 Parameter settings

Methods	Parameters						
TextSpamDetector	Datasets	Number of Filters	Window Size	AF- CNN	AF-BiLSTM	Drop out ratio	Optimizer
	SMS	128	3	tanh	ReLu	0.2	Adam optimizer with learning rate 0.001
	Twitter	128	2	Sigmoid	ReLu	0.2	
	YouTube	64	5	tanh	ReLu	0.3	
C-LSTM	Filter length = 3, Number of filters = 150, Dropout ratio = 0.5, AF = ReLu, Optimizer = RMSprop						
DECNN	Filter length = {3.4.5.6}, Number of filters = 100, dropout rate = 0.5, AF = ReLu, Optimizer = Adadelta						
DBB-RDNN-Rel	Feature maps = {10, 20, 50, 100, 200}, Learning rate = {0.05,0.10}, Input layer dropout rate = 0.2, Hidden layer dropout rate = 0.5						
SSCL	Filter length = 5, Number of filters = 128, Dropout = 0.1, AF-CNN = ReLu, AF-LSTM = Sigmoid, Optimizer = Adagrad						

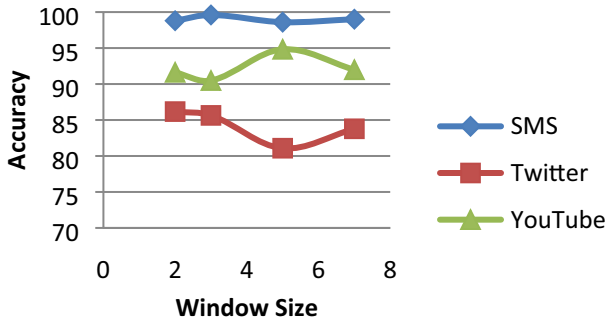

Fig. 3 Effects of window size

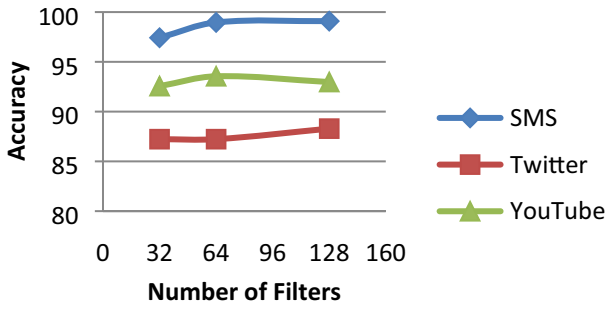

Fig. 4 Effects of number of filters

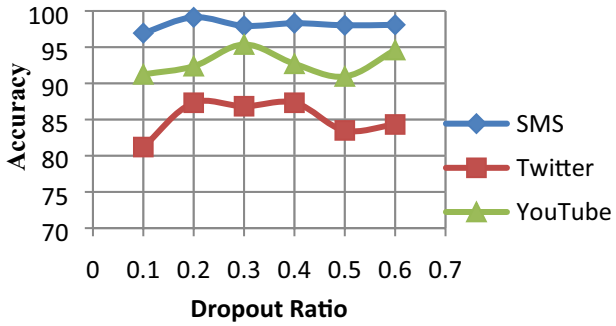

Fig. 5 Effects of dropout ratio

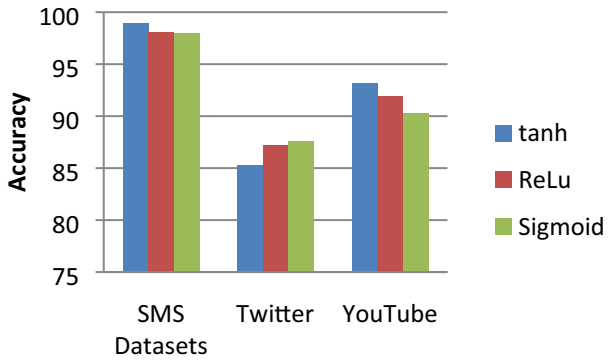

Fig. 6 Effects of activation function for CNN

Simple Attention (Bahdanau et al. 2014)—The weights for important words in the target vector are assigned by comparing the context vector generated from the input text.

Global Attention (Luong et al. 2015)—Context vector derived from all the hidden states of the encoder to attending the entire input state sequence.

Self Attention (Vaswani et al. 2017)—This attention mechanism is used to compute the representation of the input text by relating different positions of the same input text.

Context Attention (Feng et al. 2019)—This attention mechanism is applied in word and sentence level.

4.6.1 Parameter settings

In this section, the parameters influencing this model and their optimized values are discussed. When analyzing and tuning one parameter, the remaining parameters were kept constant at the basic configuration. The text should be converted into a vector form to feed into the mathematical model. In this work, word2vec is used with the word vector dimension of 300. CNN mostly uses the fixed size convolution filters. It is important to choose the right parameter value for fixed-size windows to produce feature maps. Feature maps are important as they are equivalent to the n-gram features. An experiment was conducted to verify the best value for this parameter and the results are depicted in Fig. 3.

As shown in Fig. 3, the classification accuracy is better in the window size w = 2 for the Twitter dataset, w = 5 for YouTube datasets, and w = 3 for the SMS spam dataset. Similarly, the number of filters n is examined with 32, 64, and 128 and Fig. 4 depicts 64 is better for the YouTube spam dataset and 128 produces better accuracy for the remaining datasets. The features selected after the convolutional layer are fed into activation function which is used to restrict the vector values in the specified range. To avoid overfitting, the dropout rate was used and the proper value of the dropout rate is important since if the value is too high, it results in under learning by the network and if it is a lower value then it has minimal effect. The dropout value is searched in the range of 0.1–0.7 in this work since most of the works of literature have chosen the drop out the value in this range. It can be seen from Fig. 5, that the dropout value 0.3 is performing well in the YouTube dataset and 0.2 is performing well for the remaining datasets in terms of accuracy. The remaining approaches use default parameter values. Activation function (AF) such as tanh, ReLU, and sigmoid are evaluated and the AF producing better result is used in the proposed method. AF for the CNN and BiLSTM units are depicted in Figs. 6 and 7, respectively. AF was applied to

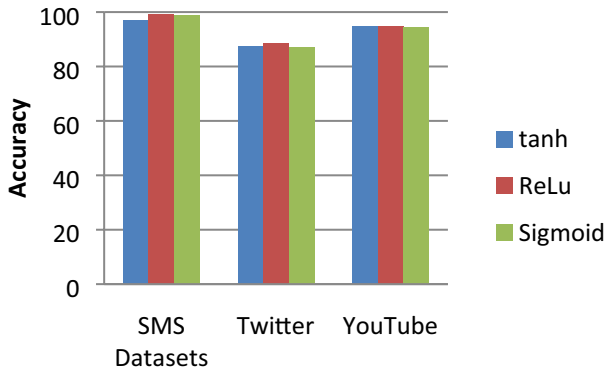

Fig. 7 Effects of activation function for BiLSTM

convolved features generated from convolution operation to limit the vector values in a certain range and this tuning is specified in Fig. 6. It is seen from Fig. 6, that sigmoid activation function is better for the Twitter dataset and tanh works well for both the SMS and YouTube datasets. ReLU activation function has performed well in BiLSTM neural network for all the datasets depicted in Fig. 7. The parameters used for the proposed model and the existing model parameters other than baseline methods are reported in Table 2.

4.6.2 Result analysis

The experimental results of the proposed approach TextSpamDetector have been compared with the baseline methods, neural network-based methods, spam detection methods, text classification methods, and existing attention mechanisms on SMS, Twitter, and YouTube datasets. The results are summarized in terms of accuracy in Table 3. From the results, we can observe that TextSpamDetector outperforms other baselines.

4.6.2.1 SMS dataset For the SMS dataset, the proposed TextSpamDetector improves the accuracy greater than 2% compared to the SVM classifier which is 0.47% only. The baseline methods have less accuracy since the methods have poor representation and highly rely on feature engineering. SVM achieves better performance than other baselines due to the optimal margin gap between separating hyperplanes which could predict better but still, it achieves lesser performance than the neural network models. TextSpamDetector stably exceeds other existing neural network methods in terms of accuracy since our model inherits the advantages from both CNN and BiLSTM and also it prevents loss of important information using a conjoint attention mechanism. When compared with the single attention mechanisms such as simple attention, self-attention, and global attention, the TextSpamDetector achieves better accuracy overcoming the attention drift by creating the representative weighted

Table 3 Accuracy comparison of the TextSpamDetector with existing approaches

Methods	Datasets		
	SMS	Twitter	YouTube
SVM	98.96	81.69	92.41
KNN	96.27	79.18	90.93
Naïve bayes	96.65	80.65	92.97
Random forest	97.42	81.12	92.89
Random tree	97.02	78.59	92.71
CNN	98.99	83.56	93.74
LSTM	98.36	83.47	93.47
BiLSTM	98.2	85.91	93.65
C-LSTM	98.71	86.47	94.37
DECNN	98.45	86.94	92.58
DBB-RDNN-Rel	98.96	87.13	94.94
SSCL	99.01	87.32	94.6
Simple attention	98.97	86.89	92.87
Global attention	98.64	86.32	94.58
Self attention	98.15	87.46	93.41
Context attention	98.27	87.17	94.02
TextSpamDetector	99.43	88.35	95.501

feature vector using different context vectors even though these single attention mechanisms are also using the deep learning techniques CNN and BiLSTM as the base model. Context attention mechanism has a lower accuracy than the proposed method even though it applies attention on different levels such as word and sentence and creates the context vector from the input text alone but the TextSpamDetector, creates the different context vectors based on different context representations generated from CNN and input text.

4.6.2.2 Twitter dataset On the Twitter dataset, the proposed TextSpamDetector achieves the best accuracy than other neural network methods and baseline methods. Specifically, TextSpamDetector gives a substantial improvement of about 8% in an average when compared to baseline classifiers which is promising, as the neural network method learns the complex structure of social text on its own. And also, TextSpamDetector is using Word2vec for word embedding which requires fewer parameters. Some existing methods DECNN, DBB-RDNN-Rel, and SSCL use different word embedding methods which require to train a large number of parameters and it causes relatively lower accuracy. It can be seen from Table 3 that, single attention mechanisms perform less in terms of accuracy compared with existing spam detection methods such as DECNN and SSCL and also with proposed TextSpamDetector. Since Twitter data are short and noisy, it creates attention drift that cannot be handled by single attention mechanisms. But TextSpamDetector incorporates two attention mechanisms to preserve the semantics

without information loss and also performs well compared to all other single attention mechanisms, context attention, and existing spam detection methods.

4.6.2.3 YouTube dataset The experimental results reported in Table 3 clearly show that the TextSpamDetector performs well in terms of accuracy when compared to all other existing methods for the YouTube dataset. YouTube comments are short in length and cannot provide statistical information for traditional methods to achieve better performance but it can be handled well by neural network methods. Especially, the high performance of TextSpamDetector shows that CNN and BiLSTM with conjoint attention mechanism have higher implications for the performance of the proposed approach. In the proposed model, CNN provides the local semantic information and BiLSTM retains the high dependency semantic information as it considers the word dependency in both the directions. Single attention mechanisms utilize this semantic information but the effect of some useful information may be flattened and they may add noise to the text representation. But, this high rich semantic information is retained using the context preserving attention and normal attention by preventing information loss in the proposed model. TextSpamDetector achieves better accuracy than the context attention as the proposed model works well on the short text datasets as it considers the different levels of context vector for attention mechanism.

4.6.2.4 F measure The F Measure comparison results of the three datasets SMS, Twitter, and YouTube are depicted in Figs. 8, 9, and 10, respectively for better illustrating the proposed model. According to Fig. 8, the proposed method has a higher F Measure than all other existing methods since

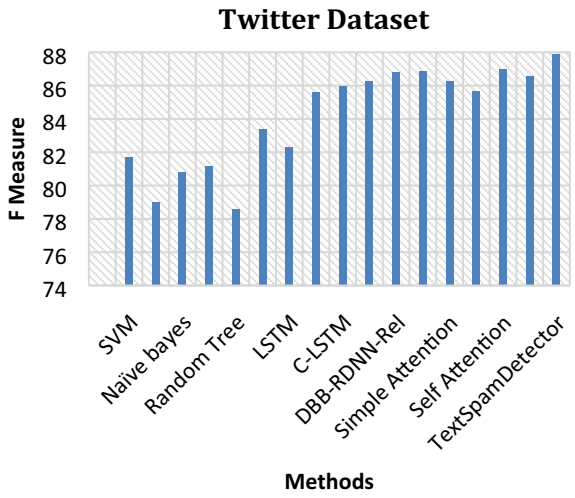

Fig. 9 F measure comparison between TextSpamDetector and existing approaches for Twitter dataset

it considers local features derived from CNN and global features retrieved from the BiLSTM with the context preserved attention mechanisms. But the proposed method is slightly similar to the performances with existing methods such as SVM, CNN, DBB-RDNN-Rel, and SSCL since the SMS dataset contains the sentences with not many special symbols and clear distinctive words which is not the real case in social network sites. From Fig. 9, it is seen that the F Measure value of the proposed method is significantly better than all other existing methods as CNN filters out the noise and BiLSTM finds the semantic sentence dependency with the weighted sensory words representation created from the proposed conjoint attention mechanism. It can be seen from Fig. 10, that the TextSpamDetector has a higher F Measure

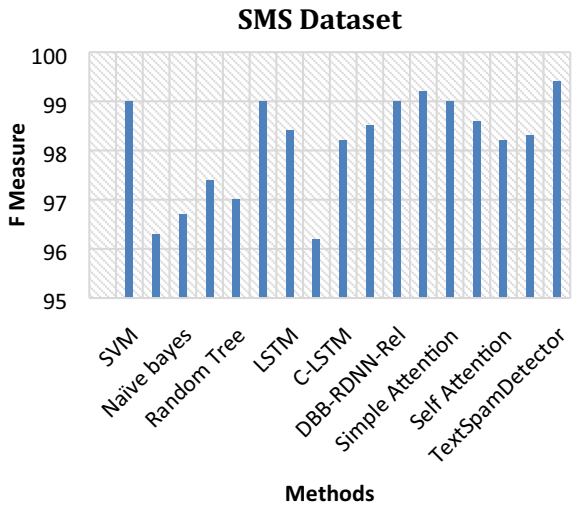

Fig. 8 F measure comparison between TextSpamDetector and existing approaches for SMS dataset

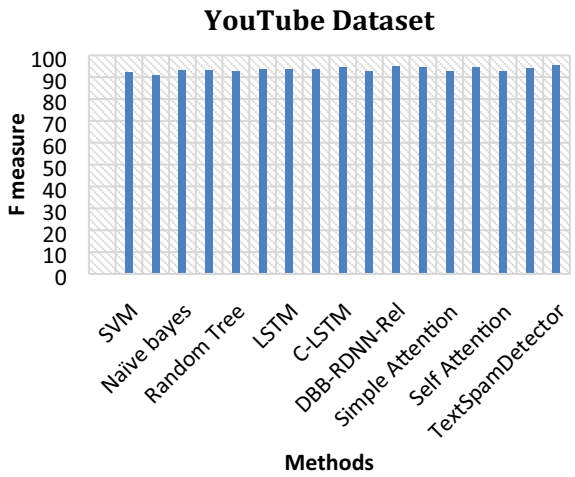

Fig. 10 F measure comparison between TextSpamDetector and existing approaches for YouTube dataset

than all other existing methods and it indicates that the TextSpamDetector has better classification performance.

4.6.2.5 False positive rate The results of the models on datasets SMS, Twitter, and YouTube are shown in Figs. 11, 12, and 13, respectively in terms of false-positive rate. The proposed TextSpamDetector has performed better as it has a slightly lower false-positive rate than all other existing methods except for BiLSTM. BiLSTM and the proposed TextSpamDetector have only a marginal difference (about 0.001%) in the false-positive rate. Social network sites are not used for official communication medium like email. If the non-spam is predicted as spam it will not lead to any major loss whereas spam being predicted as non-spam may lead to high financial loss. So the accuracy of the proposed method has a much higher impact on social network sites spam detection and TextSpamDetector has very high accuracy based on the results as listed in Table 3. However, the TextSpamDetector has a lower false-positive rate than all other existing methods for the Twitter and YouTube datasets as shown in Figs. 12 and 13, respectively.

It is confirmed by our study, that the TextSpamDetector is effective and results in a significant improvement in social network spam detection. Based on the evidence, the proposed model with the new framework can be used to achieve better performance in detecting spams with short text.

5 Conclusion

In this work, the conjoint attention mechanism is proposed to capture the comprehensive information which comprises of local semantic information, and long dependency information features without attention drift. It firstly utilizes CNN layers to extract context features and secondly uses BiLSTM to find the preceding and succeeding contextual information that represents the actual semantics of the sentence. Conjoint attention mechanism has normal attention and context preserving attention to avoid the attention drift problem by using the different level context representations for attending the important words. Normal attention has been used to focus on important information by applying the attention to the sum of hidden state representation of BiLSTM output. Context preserving attention has been applied between local representations generated by CNN after max pooling and hidden state representation of BiLSTM to retain the source semantics. The experiments were conducted on three datasets and compared with existing spam detection and text classification methods. Further, the proposed approach

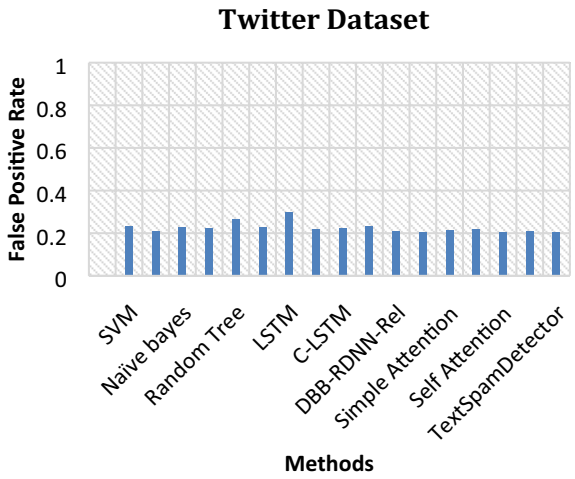

Fig. 12 False positive rate comparison between TextSpamDetector and existing approaches for Twitter dataset

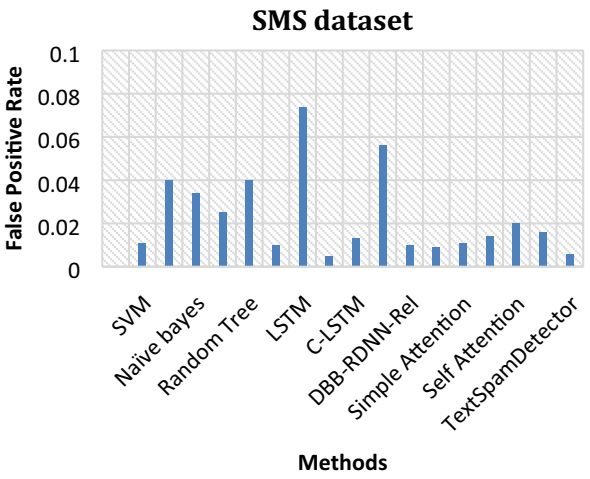

Fig. 11 False Positive Rate comparison between TextSpamDetector and existing approaches for SMS dataset

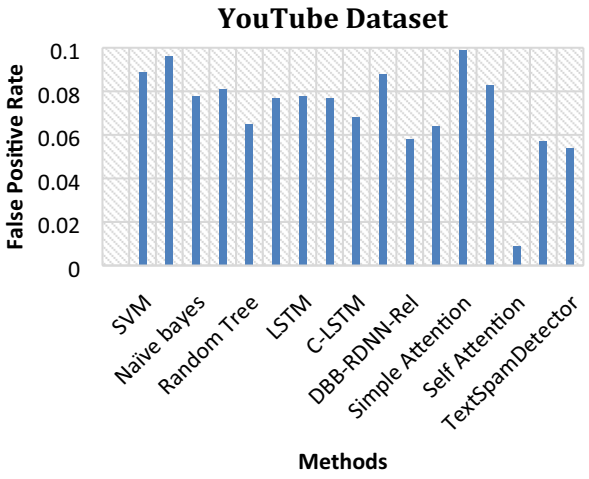

Fig. 13 False positive rate comparison between TextSpamDetector and existing approaches for YouTube dataset

tunes the parameters that show better performance in terms of accuracy. The experimental results demonstrate that the proposed method achieves better accuracy and lower false positive rate when compared to the existing methods.

References

Ahmed F, Abulaish M (2013) A generic statistical approach for spam detection in online social networks. Comput Commun 3610–11:1120–1129

Ala'M AZ et al (2018) Evolving support vector machines using whale optimization algorithm for spam profiles detection on online social networks in different lingual contexts. Knowl Based Syst 153:91–104

Alghamdi B, Watson J, Xu Y (2016) Toward detecting malicious links in online social networks through user behavior. In: 2016 IEEE/WIC/ACM international conference on web intelligence workshops (WIW)

Almeida TA et al (2016) Text normalization and semantic indexing to enhance instant messaging and SMS spam filtering. Knowl Based Syst 108:25–32

Bahdanau D, Cho K, Bengio Y (2014) Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv :1409.0473

Banerjee I et al (2019) Comparative effectiveness of convolutional neural network (CNN) and recurrent neural network (RNN) architectures for radiology text report classification. Artif Intell Med 97:79–88

Barushka A, Hajek P (2018) Spam filtering using integrated distribution-based balancing approach and regularized deep neural networks. Appl Intell 48(10):3538–3556

Benevenuto F et al. (2010) Detecting spammers on twitter. In: Collaboration, electronic messaging, anti-abuse and spam conference (CEAS), vol 6

Cao C, Caverlee J (2014) Behavioral detection of spam URL sharing: posting patterns versus click patterns. In: 2014 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM 2014)

Chen C et al (2015) A performance evaluation of machine learning-based streaming spam tweets detection. IEEE Trans Comput Soc Syst 2(3):65–76

Chen C et al (2016) Statistical features-based real-time detection of drifted twitter spam. IEEE Trans Inf Forensics Secur 12(4):914–925

Cheng Z, Bai F, Xu Y, Zheng G, Pu S, Zhou S (2017) Focusing attention: towards accurate text recognition in natural images. In Proceedings of the IEEE international conference on computer vision, pp 5076–5084

Conneau A et al. (2017) Supervised learning of universal sentence representations from natural language inference data. arXiv preprint arXiv:1705.02364

Egele M et al. (2015) Towards detecting compromised accounts on social networks. IEEE

Feng B et al (2018) Multistage and elastic spam detection in mobile social networks through deep learning. IEEE Network 32(4):15–21

Feng S, Wang Y, Liu L, Wang D, Yu G (2019) Attention based hierarchical LSTM network for context-aware microblog sentiment classification. World Wide Web 22(1):59–81

Jain G, Sharma M, Agarwal B (2019) Spam detection in social media using convolutional and long short term memory neural network. Ann Math Artif Intell 85(1):21–44

Jose T, Babu SS (2019) Detecting spammers on social network through clustering technique. J Ambient Intell Humaniz Comput. https://doi.org/10.1007/s12652-019-01541-6

Kim Y (2014) Convolutional neural networks for sentence classification. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1746–1751, Doha, Qatar, Association for Computational Linguistics

Liu S et al (2017) Addressing the class imbalance problem in twitter spam detection using ensemble learning. Comput Secur 69:35–49

Luong MT, Pham H, Manning CD (2015) Effective approaches to attention-based neural machine translation. arXiv preprint arXiv :1508.04025

Madisetty S, Desarkar MS (2018) A neural network-based ensemble approach for spam detection in Twitter. IEEE Trans Comput Soc Syst 5(4):973–984

Martinez-Romo J, Araujo L (2013) Detecting malicious tweets in trending topics using a statistical analysis of language. Expert Syst Appl 40(8):2992–3000

Mikolov T et al. (2013a) Efficient estimation of word representations in vector space. In: Proceeding of workshop at first international conference on learning representation (ICLR)

Mikolov T, Yih W, Zweig G (2013b) Linguistic regularities in continuous space word representations. In: Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: human language technologies

Rao G et al (2018) LSTM with sentence representations for document-level sentiment classification. Neurocomputing 308:49–57

Rathore S, Loia V, Park JH (2018) SpamSpotter: an efficient spammer detection framework based on intelligent decision support system on facebook. Appl Soft Comput 67:920–932

Sarıgül M, Ozyildirim BM, Avci M (2019) Differential convolutional neural network. Neural Networks 116:279–287

Sedhai S, Sun A (2015) Hspam14: a collection of 14 million tweets for hashtag-oriented spam research. In: Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval.

Sedhai S, Sun A (2017) Semi-supervised spam detection in Twitter stream. IEEE Trans Comput Soc Syst 5(1):169–175

Shehnepoor S et al (2017) NetSpam: a network-based spam detection framework for reviews in online social media. IEEE Trans Inf Forensics Secur 12(7):1585–1595

Simon K (2020) Digital 2020: 3.8 billion people use social Media. We Are Social Inc. https://wearesocial.com/blog/2020/01/digital-2020-3-8-billion-people-use-social-media. Accessed 20 Feb 2020

Song L et al (2017) Who are the spoilers in social media marketing? Incremental learning of latent semantics for social spam detection. Electron Commer Res 17(1):51–81

Tagg C (2009) A thesis on A corpus linguistics study of SMS text messaging. University of Birmingham, Diss

Thomas K et al (2011) Design and evaluation of a real-time URL spam filtering service. 2011 IEEE symposium on security and privacy. Trans Dependable Secure Comput 14(4):447–460

UtkMl's Twitter Spam Detection Competition (2019).https://www.kaggle.com/c/twitter-spam/data. Accessed Nov 2019

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Polosukhin I (2017) Attention is all you need. In: Advances in neural information processing systems, pp 5998–6008

Wu T et al (2017a) Detecting spamming activities in twitter based on deep-learning technique. Concurr Comput Pract Exp 29(19):e4209

Wu T et al (2017b) Twitter spam detection based on deep learning. In: Proceedings of the australasian computer science week multiconference

Xu G et al (2019a) Sentiment analysis of comment texts based on BiLSTM. IEEE Access 7:51522–51532

Xu J et al (2019b) Incorporating context-relevant concepts into convolutional neural networks for short text classification. Neurocomputing. 33:10067–10068
Yang C, Harkreader R, Guofei Gu (2013) Empirical evaluation and new design for fighting evolving twitter spammers. IEEE Trans Inf Forensics Secur 8(8):1280–1293
Zhang X et al (2016) Detecting spam and promoting campaigns in Twitter. ACM Trans Web (TWEB) 10(1):1–28
Zheng X et al (2015) Detecting spammers on social networks. Neurocomputing 159:27–34
Zhou C et al (2015) A C-LSTM neural network for text classification. arXiv preprint arXiv:1511.08630
Zhou Y, Xu B, Xu J, Yang L, Li C (2016) Compositional recurrent neural networks for chinese short text classification. In: 2016 IEEE/WIC/ACM international conference on web intelligence (WI), pp. 137–144.