Annals of Mathematics and Artificial Intelligence (2019) 85: 21–44
https://doi.org/10.1007/s10472-018-9612-z

Spam detection in social media using convolutional and long short term memory neural network

Gauri Jain[1] · **Manisha Sharma**[1] · **Basant Agarwal**[2]

Published online: 2 January 2019

Abstract
As the use of the Internet is increasing, people are connected virtually using social media platforms such as text messages, Facebook, Twitter, etc. This has led to increase in the spread of unsolicited messages known as spam which is used for marketing, collecting personal information, or just to offend the people. Therefore, it is crucial to have a strong spam detection architecture that could prevent these types of messages. Spam detection in noisy platform such as Twitter is still a problem due to short text and high variability in the language used in social media. In this paper, we propose a novel deep learning architecture based on *Convolutional Neural Network (CNN)* and *Long Short Term Neural Network (LSTM)*. The model is supported by introducing the semantic information in representation of the words with the help of knowledge-bases such as *WordNet* and *ConceptNet*. Use of these knowledge-bases improves the performance by providing better semantic vector representation of testing words which earlier were having random value due to not seen in the training. Proposed Experimental results on two benchmark datasets show the effectiveness of the proposed approach with respect to the accuracy and F1-score.

Keywords Spam detection · Deep learning · Sequential Stacked CNN-LSTM · CNN · LSTM

1 Introduction

Social Media has come up as the most popular means for people to communicate globally. During the last few years it has evolved a great deal (from a monologue driven blogs to

✉ Gauri Jain
jain.gauri@gmail.com

Manisha Sharma
manishasharma8@gmail.com

Basant Agarwal
basant@skit.ac.in

[1] Department of Computer Science, Banasthali Vidyapith, Banasthali, India

[2] Department of Computer Science and Engineering, Swami Keshvanand Institute of Technology, Jaipur, India

social media networks) and has become far more interactive, dynamic and different than the traditional media like TV, movies and newspapers etc [13]. Starting with SMS through mobile phones to popular social networking platforms like Facebook, Twitter, Instagram,[1] Reddit[2] and other web based services, the focus has now moved from web based applications on laptops to App based services on mobile phones. One of the most trending social media websites is Twitter. Initially, Twitter was introduced as a group SMS service in 2006, which was used to send short messages known as "Tweet" to all the friends in the defined network. However, now it is widely used by people to share the daily status with a broad range of listeners or audience [18].

The increase in user base of Apps like Twitter has led to the generation and exchange of huge volume of date on the web [10]. While the power to rapidly share information attracts individuals and companies, it also acts as an attraction force for people sending unwanted and unsolicited messages over the network. This type of message is known as a *spam* message. They are generally marketing, fraud or offensive messages that try to take advantage from the receivers. Spam Detection started with manual filtering of messages, followed by simple filtering rules that could detect a message with some known properties. Automatic spam detection started with the use of traditional machine learning methods that were used to create spam detection model.

In traditional spam detection, simple techniques such as blacklisting and content-based machine learning methods are often used for spam filtering [26]. Existing such methods performed quite well on longer email messages, but, in recent times, spam detection in short and noisy platform further increased the challenges in spam detection [42]. In short text such as in Twitter and SMS domain, spam detection is more challenging problem due to noisy and small size of these messages. The following tweet is an example of the same: *"RT @Stormzy1: The clean hearted alwayssssss win in d end. u badmind lil weirdos wid u r bad energies are gonna destroy urselves trust"*, another example, *"Aft i finish my lunch then i go str down lor. Ard 3 smth lor. U finish ur lunch already?"*. Here, many words are mis-spelled and tweets are very noisy. To deal with such kind of messages is very challenging. Furthermore, these messages are usually full of slangs, symbols, emoticons, mis-spelled words, abbreviations, and irregular grammar that make it difficult to develop a reliable spam detection model. The existing successful methods don't perform well on these kinds of short and noisy messages. In addition to these problem, there is also a problem of presence of ambiguous words in the text that harm the performance of existing traditional machine learning techniques, which we tackle by using semantic representation of words [29].

The latest development in the field of classification is through Deep Learning Technologies. They are known to demonstrate a remarkable performance in the field of Natural Language Processing (NLP) [19]. In this paper, a novel architecture is introduced that combines *Convolutional Neural Network (CNN)* and *Long Short Term Memory (LSTM)* neural language models and proposed a hybrid deep learning architecture which we have named as *Sequential Stacked CNN-LSTM model (SSCL)* for the spam classification. It offers more diverse text representation which is further enhanced by training the network on the top of pre-trained vectors. In the *SSCL* model, the text is converted to the vector form with the help of *word2vec*. This process is also supported by the use of semantic dictionaries like *WordNet* and *ConceptNet*. These dictionaries help in extracting the closest semantic word for a given word, for which no corresponding word vector could be found using *word2vec*. CNN

[1] www.instagram.com

[2] www.reddit.com

perform the task of extracting the most important *n-gram* features from the text sentence and in the following layer LSTM works upon these features sequences by capturing long-term dependencies. The proposed architecture is evaluated on spam classification tasks and compared with traditional machine learning models as well as individual CNN and LSTM architectures. The work presented here is an extension of our previous work [14] and [15]. The experimental results show that SSCL architecture produces better results when compared with several benchmark models as well as the CNN and LSTM models when used individually.

The main contributions of this paper are as follows:

1. We proposed a novel deep neural architecture combing *CNN* and *LSTM*. In the proposed approach, we include semantic information in sentence representation.
2. We present an empirical study to select the hyperparameters for the deep neural architecture. We provide insights in developing a robust and reliable model for spam detection.
3. We propose to use the knowledge-bases *WordNet* and *ConceptNet* to improve the coverage of the words which are available in the testing set but not seen in the training dataset. It could improve the performance of the proposed model because proposed model could learn the domain-specific embeddings for more words with the task-specific training data.
4. As better initialization of word vectors has been proved to be useful for improvement of the performance of deep learning method, we propose a novel way of enriched word vector representation. We show with extensive experimental results that by using *WordNet* and *ConceptNet* to improve the coverage of the words, the performance of the spam detection improves.
5. We enhance the word representation by leveraging both the training data as well as the knowledge-base systems.
6. We also present an extensive study of the related work done in the area of spam detection for social media and specially for short-text messages.

The organisation of the paper is as follows. Section 2 presents the problem statement addresses in this paper. Section 3 presents the related work. Section 4 gives the detail about the proposed SSCL approach that includes two main stream algorithms: CNN and LSTM along with the different embedding techniques. Section 5 is the experiments and results section with error analysis. Section 6 concludes the paper with summarizing the contributions.

2 Problem definition

Spam detection is considered as a NLP classification problem using machine learning algorithms. The problem of spam detection addressed in this paper is described as follows. Given a collection of M annotated text messages $\{x_1, x_2, x_3 ... x_n\}$, for ($i = 1$ to M) with annotations m_i. The annotations m_i indicate the i text message is a *spam ham* message. Spam detection is considered a classification problem, in which for a given short text message, the objective is to classify as *spam* or *ham*. The problem statement is to develop a robust and reliable spam detection model which can determine a given message as spam or ham.

For example tweet message from the corpus are as follows:

1. *"Want to double, even triple your income? Listen Live: https://t.co/xiys3MPqh6 Leading With Purpose TalkRadio"*

2. *"Sorry man my account's dry or I would, if you want we could trade back half or I could buy some shit with my credit card."*

Here, the objective is to determine if these given sentences are *spam* or *ham* message. In literature, many successful spam detection approaches have been proposed mainly for long email messages, but to determine if a message is spam or ham for short and noisy text is still quite challenging. In this paper, we focus on developing a reliable deep learning based spam detection model which can efficiently determine if a given short and noisy text is a spam or ham.

3 Related work

Spam is an unsolicited and unwanted message that is sent through an electronic medium. There are different types of social media through which the spam is sent or received. Proposed work is related to two directions of research (i) Spam detection using supervised methods, and (ii) Advancements in Deep learning techniques for NLP applications specially for spam detection.

3.1 Spam detection using supervised methods

Initially, spams started spreading with email spam known as unsolicited bulk email (UBE) or unsolicited commercial email (UCE). Further, the SMS being a very cost effective method used for sending individual messages to the prospective clients, has a higher response rate as compared to email spam. Along with emails and SMS, social networking like Twitter [10], Facebook, instant messenger like WhatsApp etc. are also contributing to a major chunk of spam over the network. Spam detection is a tedious task if no automatic filter is installed at the receiving end. One of the initial classifiers is rule based filtering in which the rules are more formally written and can be deployed to a wide area of clients. It includes a set of pre–defined rules that are applied to an incoming message and the message is marked as spam if the score of the test exceeds the threshold specified. Survey of anti-spam tools is provided by Stern et al. [35] and Cournane et al. [6]. Many companies have additional checks in the form of white–listing, black–listing [18, 38] and grey-listing [23] and the source of incoming message is cross checked and marked spam based on the list of these URLs. Martinez et al. [25] detected spam tweets by exploiting the difference between the language model of the tweet and the page linked with that tweet. However, the success of these methods is limited and they need to be combined with other machine learning methods in order to give fairly good results. Some of the most common classifiers used for spam detection are SVM, Naive Bayes, Artificial Neural Network, and Random Forests. These classifiers require a sophisticated way to extract features from the text to learn the pattern of spam and ham messages. The most common model for feature extraction is Bag-of-words (BoW). There are different weighing schemes in BoW model like Term Frequency (TF), Inverse Document Frequency (TF-IDF) etc., but all of them uses the token frequency in some form.

Naive Bayes is the most effective and a simple statistical classifier. It is most widely used for spam detection, it assumes that the features extracted from the word vector are independent of each other. Kim et al. [17] experimented with different number of features used for spam classification using the same algorithm. Androutsopoulos et al. [3] also performed a comparison between Naive Bayes and key–word based spam filtering on social

bookmarking system and concluded that the performance of Naive Bayes is better amongst the two. Almeida et al. [2] used different techniques like document frequency, information gain, etc. for term selection and used it with four different versions of Naive Bayes for spam filtering. He concluded that Boolean attributes perform better than others and MV Bernoulli performs best with this technique. Like Naive Bayes, SVM is also used for detection of spams from various social media like Twitter [26], Blogs [20] etc. There are various variations introduced to further enhance the performance of SVM classifier. For e.g. Wang et al. [40] proposed GA-SVM algorithm in which genetic algorithm used for feature selection and SVM for the classification of spams and its performance was better than SVM. Tseng et al. [39] proposed an approach that showed an incremental support to SVM by extracting features from the users in the network. This model proved to be effective for the detection of spam on email. Functional spam detection is done with the help of temporal position in the multi–dimensional space. Other than SVM, another functional classifier is k–NN [12]. Artificial Neural Network (ANN) has also shown promising results in the area of spam detection. Sabri et al. [31] used ANN for spam detection in which the useless input layers could be changed over a period of time with the useful one. Silva et al. [33] compared different types of ANN like MLP, SOM, Levenberg-Marquardt algorithm, RBF for content based spam detection, concluded that they are effective in the detection of spam. Ensemble methods like random forest have also proven their capability as an effective classifier. DeBarr et al. [7] used clustering along with Random Forest for spam classification. Several researchers provides the comparison between the above discussed classifiers in literature [16, 44]. The spam detection using baseline methods are summarized in Table 1.

Table 1 Baseline machine learning methods

Paper	Method/Technology	Dataset	Remarks
Androutsopoulos et al. [3]	Naive Bayes + Keywords	Social Bookmarking	Outperforms Naive Bayes
Zhang et al. [43]	SVM, AdaBoost, Maximum Entropy, Naive Bayes	PU1 Corpus, Ling Spam Corpus, SpamAssassin, ZH1 Chinese Spam Corpus	SVM, AdaBoost, Maximum Entropy outperforms
Wang et al. [40]	GA + SVM	Spam base email Dataset	GV-SVM outperforms SVM
Kolari et al. [20]	Various content features + SVM	Crawled Blogs	Words + url + SVM outperforms
Kim et al. [17]	Naive Bayes + Keywords	Social Bookmarking	Outperforms Naive Bayes
Tseng et al. [39]	User based features + SVM	Live email from server	Outperforms SVM
Sabri et al. [31]	ANN	SpamAssassin Corpus	0.534% FP and 3.66% FN
Silva et al. [33]	ANN	WEBSPAM-UK 2006	Outperforms SVM and decision trees
Wu et al. [41]	User based + content based features	Sina Weibo account	Outperforms SVM, co-classification, Logistic regression, Least square methods

Table 2 Twitter spam detection

Paper	Method/Technology	Dataset	Remarks
Stringhini [36]	Social behavior of spammers + SVM	Twitter Accounts	Identified 15,857 spam Twitter profiles
Mccord et al. [26]	User based features + RF/ NB/ SVM/ KNN	Crawled Tweets	User based features + RF outperforms
Miller et al. [29]	Content based features	Twitter Accounts	Recall 100% and FP 2.2%

Work in the area of spam and spammer detection on Twitter is already on progressive path. Miller et al. [29] used the content based features like hashtags, mentions etc along with twitter stream clustering methods to identify spammers. Wu et al. [41] used a unified approach consisting of spammer and spams to refine the spam detection results.The same approach was followed by Stringhini et al. [36] in which the social behavior of spammers was combine with the content based features along with SVM to detect the spammers on Twitter. The work on Twitter spam detection is summarized in Table 2.

3.2 Deep learning techniques for spam detection

Deep Learning Models have proven their capabilities in the area of NLP tasks like distributed word learning, sentence and document representation [27], parsing [34], statistical machine translation [8], sentiment classification [1, 19], etc. External domain knowledge is important in case of sentence representation through NN in NLP text classification. In many recent researches, the input word sequence and syntactic parse tree are used for creating the text representations while modelling RNN and CNN. The pooling feature of CNN add power to the model by capturing local as well as high-level features from consecutive context windows. Collobert et al. [5] applied max pooling operation to successive windows to extract global features. Kim [19] also proposed a CNN architecture with a number of filters with various window size and two different 'channels' of word vectors. Tao et al. [21] applied tensor-based operations on concatenated word vectors in the convolutional layer. Mou et al. [30] applied convolutional models on sentences having hierarchical structures. Many variants of RNN have been proposed which is able to handle variable length input sequences [4]. Tai et al. [37] changed LSTM structure to model tree-structured topologies by stacking both CNN and LSTM in a sequential manner and achieved promising results for semantic sentence modeling. This combination of two deep learning structures can also be seen in some computer vision tasks like image caption [21] and speech recognition [32]. Most of these models use multi-layer CNNs or train CNNs and RNNs separately or throw the output of a fully connected layer of CNN into RNN as inputs. Lei et al. [22] in his research proved that sequential models are able to capture the structural semantics for NLP, thus he built the CNN model on word sequences. Recently, Wu et al. [42] proposed a deep learning based approach to detect the spam messages on Twitter. Their approach is based on different syntax analysis, blacklist analysis and feature analysis in addition with *word2vec* based features. A work related to spammer detection is detecting rumors from the microblogging websites: Twitter and Weibo using LSTM and GRU by Jing et al. [24]. Gao et al. [9] used RNN for its simple advantage that the inputs can of variable length, to detect the email spam. Our work on spam detection clearly suggests the efficiency of our semantic model over individual CNN or LSTM deep learning model and as well as traditional

Table 3 Deep learning techniques for spam detection

Paper	Method/Technology	Dataset	Remarks
Gao et al. [9]	RNN	Email	Overcome the disadvantage of fixed length feature vector and results were better than baseline methods
Ma et al. [24]	LSTM and GRU	Twitter	Outperforms baseline methods with extra hidden layers in lesser time
Wu et al. [42]	Word2vec, doc2vec, softmax classifier	Twitter	Better performance than text based and non-text based baseline methods

classification models. The summary of the spam detection using deep learning techniques are summarized in Table 3.

4 Proposed deep learning based models for spam detection

In this research, our focus is on spam classification in social media using deep learning based approaches. The main concept of the paper is that we add new semantic layer just before the embedding layer that incorporates semantic information. We present three different architectures to achieve this, firstly, we use semantic layer with convolutional neural network, we call this model as Semantic Convolutional Neural Network (SCNN). Secondly, we use the semantic layer with the LSTM neural network, this model is named as Semantic long short term memory (SLSTM). Finally, we present a hybrid of these two approaches, in which we combine the CNN and LSTM to make final predictions, we name this hybrid model as Sequential Stacked CNN-LSTM Model (SSCL). The details of these models are desbribed in subsequent subsections.

4.1 Semantic convolutional neural network (SCNN)

SCNN is a convolutional neural network with an additional layer into its semantic representation. The initial text along with its *word2vec* based representation is enhanced using *WordNet* and *ConceptNet* knowledge-bases, and further, fed into the traditional convolutional neural network. The details on how we enhance the semantic representation is explained in detail in the Section 4.3.

4.2 Semantic long short term memory (SLSTM)

The LSTM is a variant of recurrent neural network with an enhanced semantic representation of the words. The only difference with the SCNN model is that in this approach, we use LSTM neural network in place of CNN network to evaluate the robustness of the proposed approach. Further, we also present an approach which combines both of these two approaches SCNN and SLSTM that we explain in next subsection.

4.3 Sequential Stacked CNN-LSTM Model (SSCL)

Proposed deep learning architecture i.e. *Sequential Stacked CNN-LSTM (SSCL)* model combines CNN and LSTM neural architecture along with the semantic model to construct an

end to end architecture for spam detection in social media. Proposed *SSCL* neural architecture take advantage of both CNN and LSTM architectures. The *SSCL* model extracts the local-region features and *n-gram* information using CNN neural network and long-term dependency information using LSTM network. In addition, the semantic information is introduced in the word representations with the help of *Word2Vec* Embeddings, *WordNet*, and *ConceptNet*, which further improved the performance of the spam classification task. The architecture of the Sequential Stacked CNN-LSTM (SSCL) model is shown in Fig. 1. The figure shows the 3 layers (semantic, CNN and LSTM) sequentially stacked over each other. In the *SSCL* approach, a single CNN layer is used on the text data that extract the feature sequences for windows. These features sequences are directly used by LSTM to learn long-range dependencies and finally sigmoid function produces the final classification labels as *spam* or *ham*. The detail working of the SSCL model is described in Algorithm 1 and also demonstrated in Fig. 2.

4.3.1 Embeddings

For classifying the text, they have to be in the numerical form which is mostly carried out by assigning rando m vectors with real number values. These vectors do not show any relation with each other. In this paper, we have used Word2vec model to convert the text data into numerical vectors representing words. If the corresponding does not exist in Word2vec embeddings, then a similar word is identified using the WordNet and ConceptNet semantic dictionaries. The word vectors are mapped in the multidimensional space and the distance between semantically similar words is lesser as compared to words having no relation which adds semantic meaning to these data representation.

i. *Word2Vec*

Word2vec is based on multilayer neural network proposed by Tomas Mikolov [27] and his team at Google. It provides word vectors for textual words, also known as word embeddings. It takes textual words as an input and the corresponding word vectors as an output. It aims at grouping the vectors of the similar words nearer to each

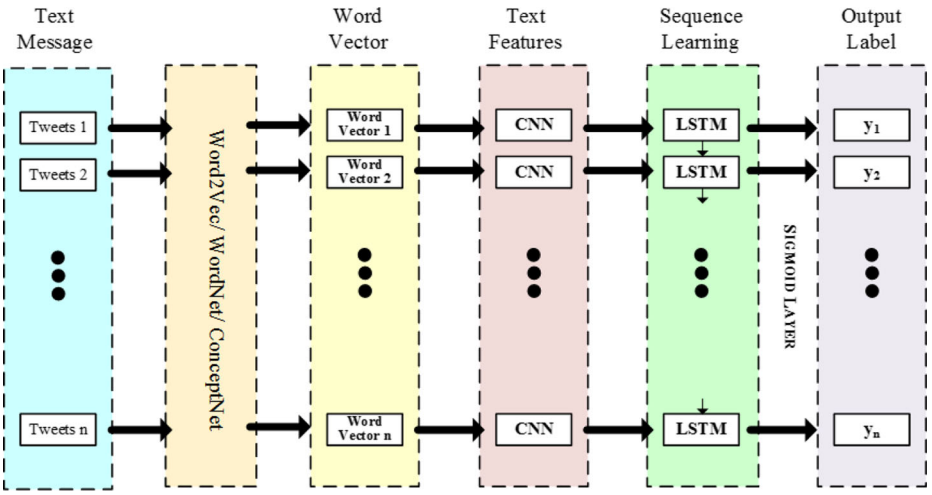

Fig. 1 The sequential stacked CNN-LSTM architecture

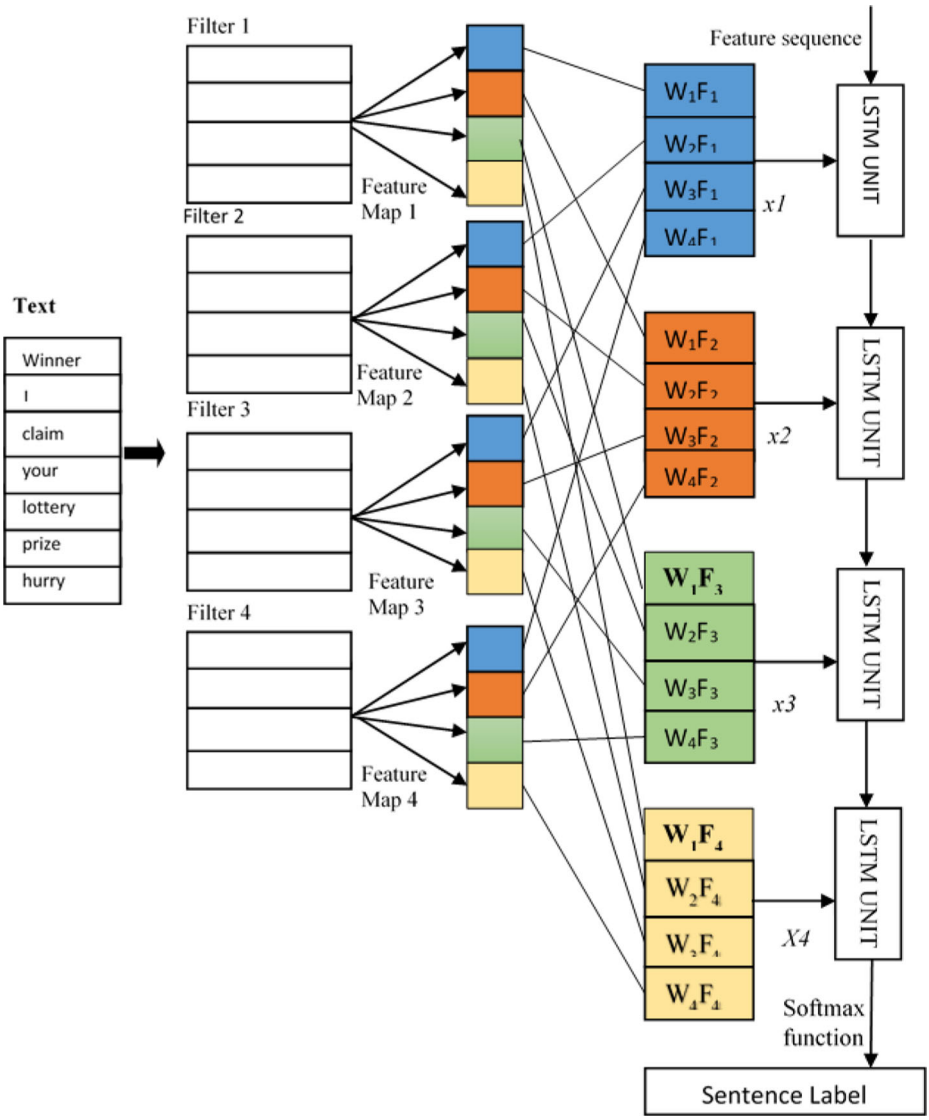

Fig. 2 The overview of the proposed SSCL model

other while the unrelated words farther in the multidimensional space mathematically without any human intervention. We have used pre–trained vectors from Google's Word2Vec having dimension 300 to convert the text into vector form.

ii. *WordNet* WordNet [28] is a large database of lexical words belonging to English language resembling a thesaurus. In this dictionary, the words related to each are grouped together into a set known as Synsets. Lexical categories like nouns, verbs, adjectives, etc. form different synsets and they are connected via conceptual-semantic relations and lexical relations. This is a sort of dictionary and thesaurus that can gives us word meanings along with associations with other words. Therefore, words are like nodes

Algorithm 1 SSCL algorithm

Input: *Text Sentence*
Output: *Text Label: Spam or Ham*

for *each text sentence* $s \in \{1, ..., N\}$ **do** *//convert text into numerical word vectors*
  if $Word2Vec(s_n) \neq NULL$ **then**
    $x_n = Word2Vec(s_n)$
  else if $WordNet(s_n) \neq NULL$ **then**
    $s_n = WordNet(s_n)$
    $x_n = Word2Vec(s_n)$
  else if $ConceptNet(s_n) \neq NULL$ **then**
    $s_n = ConceptNet(s_n)$
    $x_n = Word2Vec(s_n)$
  else
    $x_n = Random(s_n)$
  end if
end for
for *each filter* $f \in \{1, ..., F\}$ **do** *//Get the most important features*
  $c = CNN(x)$
end for
for *each time step* $t \in \{1, ..., T\}$ **do**
  $o = LSTM(c)$
end for
for *each sentence representation step* $o \in \{1, ..., N\}$ **do** *//Get the final sentence label*
  $label = sigmoid(o)$
end for

and the connections represents the associations between them. For E.g., the synset for the word spam is shown in Table 4 [28].

iii. *ConceptNet*

ConceptNet is a collection of semantic concepts used in our day to day life [11]. These common sense concepts are taken by the interactions of the common people over Internet. It is the largest publically available *common–sense knowledge-base* consisting of more than 250,000 relations. It can be used to gain knowledge as it extract the inferences from the text documents. The nodes are the concepts and the connections are the relations between the nodes. Some of the most common relationships

Algorithm 2 CNN algorithm

Input: *Sentence Matrix x(L* $\times$ *d), F filters*
Output: *Most Important Features*

for *each filter* $f \in \{1, ..., F\}$ **do** *//Get the most important features*
  $w_j = [x_j + x_{j+1} + ... + x_{j+k-1}]$
  $c_j = ReLU(w_j n + b)$
  $c = (c_1 \oplus c_2 \oplus ... \oplus c_j)$
end for

Table 4 Words and Synset

Word	Synset
spam(n)	junk e-mail, spam - unwanted e-mail (usually of commercial nature sent out of bulk)
	spam - a canned meat made largely from pork
spam(v)	spam - send unwanted or junk e-mail
content(n)	content, depicted object, subject - something (a person or object or scene) selected by an artist or photographer for graphic representation
	cognitive content, content, mental object - the sum or range of what has been perceived, discovered, or learned
	content, message, subject matter, substance - what a communication that is about something is about
	content, contents - (usually plural) everything that is included in a collection and that is held or included in something
	capacity, content - the amount that can be contained
	content - the proportion of a substance that is contained in a mixture or alloy etc.
	content, contentedness - the state of being contented with your situation in life
content(v)	content - satisfy in a limited way
	content - make content
content(a)	content, contented - satisfied or showing satisfaction with things as they are

between concepts are IsA, TypeOf, UsedFors, etc. For example, given two concepts "spam" and *"junk e-mail"*, an assertion between them is IsA; that shows that a spam is a junk email as shown in Fig. 3. The figure shows a sample subgraph of the ConceptNet common-sense ontology from the the word '*Spam*'.

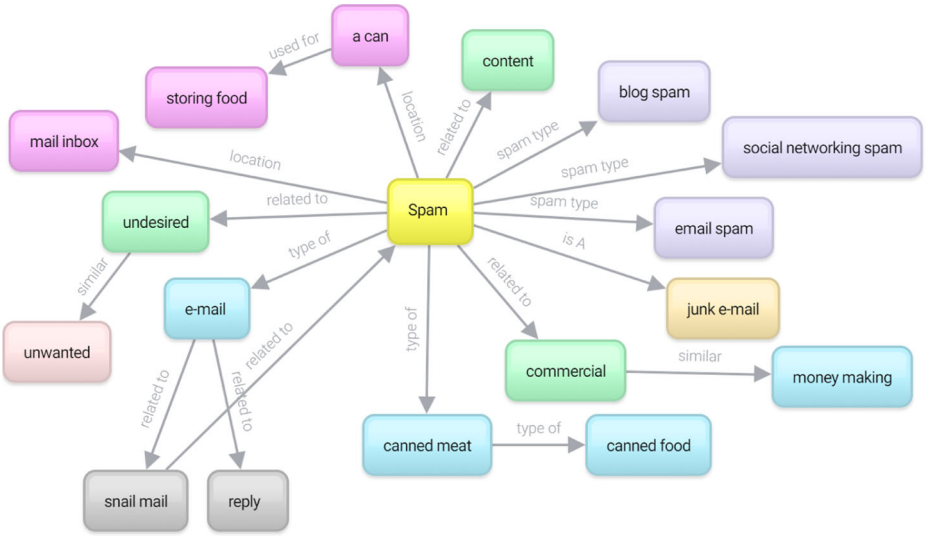

Fig. 3 Sample subgraph from ConceptNet ontology

4.3.2 Feature extraction using CNN

Feature extraction is carried by applying the convolution operation on the already converted sentences into word vectors. It involves the filter to slide over the word vectors taking a few words at a time which helps to detect local features in the filter window. This is a one-dimensional convolution as it involves taking a number of row from the sentence matrix at a time. Let x be the input sentence having L words and d is the length of a word vector. Therefore, $x \epsilon R^{L \times d}$ represents the input sentence with the length L. Therefore, any word x_i in the sentence is a d-dimensional word vector such that $x \epsilon R^d$. A filter f of length k (number of words) is chosen, where vector $n \epsilon R^{k \times d}$ represents a filter for the convolution function then for the words starting at the position j till the position $(j + k - 1)$ will be processed by the filter at a time. The window w_j is denoted by (1).

$$w_j = [x_j + x_{j+1} + ... + x_{j+k-1}] \tag{1}$$

Here, the $(+)$ operator represents the concatenation operation for word vectors. The number of filters can range from 1 to few hundreds depending on the text. The filters f convolves with a window at a time to generate a feature map for that window $c_j \epsilon R^{L-k+1}$, where, c_j is calculated as in (2)

$$c_j = f(w_j \cdot n + b) \tag{2}$$

where $\cdot$ is an element-wise multiplication, b ϵ R is a bias term and f is a nonlinear transformation function. The transformation function is *tanh*, *sigmoid*, *ReLU* etc. The feature maps for each window (with different filters) are concatenated to get a high level vector representation and fed as an input to the *LSTM* unit. Convolution operation is followed by a pooling operation which helps to select only important information by removing low activation information for further processing and avoid overfitting due to noisy text. The detailed algorithm is described in Algorithm 2.

4.3.3 Learning high level dependencies using LSTM

The traditional methods is not able to correlate the current information with the past information. Therefore, they process the current information without the knowledge of past. Long Short-Term Memory Neural Networks (LSTM), has an ability to memorize and pass the past information in a chain-like neural network architecture. RNN also passes the information learned to the next layer but as the gap between two time step increases, they suffer from the problem of vanishing/exploding gradient. This is due to the calculation of gradient during the backpropagation which results in an inability to memorize long term memory. LSTM structure has LSTM units consisting of memory cells that selectively stores the information for a longer period without getting degenerated.

The gates are not used for sending the input, instead, they are used to set the weights on the connections between the neural network and the memory cell. The memory cell also has a self-connection. When the forget gate has value 1 and the self-connection also has a weight 1, then the memory cell retains the content but with the 0 value in forget gate, the memory cell discards its contents. When the input gate is 1, the neural network is able to write into the memory cell and when the output gate is 1, the network can read the values from the memory cell. These three cells in the LSTM unit protects the network from exploding and

Table 5 Dataset description

Dataset	No. of Instances	No. of Ham	No. of Spam
SMS Spam	5574	4827	747
Twitter	5096	4231	865

vanishing gradient problem. LSTM processes the data in a sequential manner and while processing the current input, it takes into account the current input x_t as well as the output of the previous hidden state h_{t-1} at each time step. Therefore, LSTM is used for successful spam detection.

The Algorithm 3 demostrates the gates and the update operations each of the gates of a LSTM unit.

Algorithm 3 LSTM algorithm

Input: *Word Vectors*
Output: *Sentence representation*

for *each time step* $t \in \{1, ..., T\}$ **do** *//Update all the state and gates of the LSTM cell*

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$$
$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$$
$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$$
$$\check{c}_t = tanh(W_c x_t + U_c h_{t-1} + b_c)$$
$$c_t = i_t \odot \check{c}_t + f_t \odot c_{t-1}$$
$$h_t = o_t \odot tanh(c_t)$$

end for

5 Experimental settings and results

The implementation of Sequentially Stacked CNN – LSTM model was performed on the Keras 2.0 API with Tensorflow backend using Python 2.7 with Ubuntu 16.4.2 operating system.

5.1 Dataset description

For the evaluation of the *SSCL* architecture, two datasets have been used: SMS spam and Twitter dataset. First, the SMS Spam dataset, which is available in UCI repository.[3] The dataset consists of 5,574 English, and non-encoded SMS text messages which is a mix of pre-labelled 4,827 *ham* and 747 *spam* messages. The class wise SMS distribution is shown in Table 5. The training and testing instance distribution according to the class labels is shown in Table 6.

We have also used tweets from the Twitter dataset which were scraped from public live tweets. These tweets are manually labelled as *ham* or *spam* by two annotators ignoring the tweet if there is any conflict among them. The class-wise distribution is shown in the Table 5 while the training and testing classwise instance distribution is shown in Table 7.

[3] http://dcomp.sor.ufscar/talmeida/smsspamcollection

Table 6 No. of training and testing samples used in SMS Spam dataset

Number	Class	Training	Test
1	Spam	597	150
2	Ham	3862	965
	Total	4459	1115

5.2 Evaluation measures

Standard metrics like precision, recall, accuracy, and F1 score are used for evaluation purposes as shown in Table 8. The confusion matrix consists of the following measures:

1. **True Positive (TP):** A test result that detects the condition correctly when the condition is present.
2. **True Negative (TN):** A test result that does not detect the condition when the condition is absent.
3. **False Positive (FP):** A test result that detects the condition when the condition is absent.
4. **False Negative (FN):** A test result that does not detect the condition when the condition is present.

Various evaluation measures are defined below:

1. **Accuracy**: It is the number of correct predictions made divided by the total number of predictions made.

$$Accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)} \tag{3}$$

2. **Precision**: It is the number of positive predictions divided by the total number of positive class values predicted.

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

3. **Recall:** It is the number of positive predictions divided by the number of positive class values in the test data.

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

4. **F-Measure:** The F Measure conveys the balance between the precision and the recall.

$$FMeasure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{6}$$

Table 7 No. of training and testing samples used in Twitter dataset

Number	Class	Training	Test
1	Spam	672	173
2	Ham	3406	845
Total		4078	1018

Table 8 Confusion matrix

	Ham	Spam
Spam	True Positive (TP)	False Positive (FP)
Ham	False Negative (FP)	True Negative (TN)

5.3 Hyperparameters setting

Here, we provide details on the empirical analysis of optimization of hyoerparameters for the proposed deep learning based model for spam detection.

5.3.1 Word vector initialization

For text to be fed in a mathe matical model, it has to be converted into a numerical form known as word vector. This is done by initializing them with random values having a uniform distribution. In our model, the word vector of 300 dimension is initialized with the help of word embedding from Google's Word2vec [27].

5.3.2 Filter size and no. of filters

Filter size is an important parameter of convolutional neural network to consider while training. The performance of the model with two filter size 4, 5 is observed, the model performed best in 5-gram phrases with SMS spam dataset while it gave better results for *4*-gram phrases in the case of Twitter dataset as shown in Table 9. On the other hand, increasing the number of filters from 32 to 128 increases the accuracy of model on SMS Spam dataset while, best accuracy for Twitter is seen with 64 filters as shown in Table 10.

5.3.3 Activation function

The best features after the convolution layer is selected when the activation function is applied on the convolved features. Convolved features are the result of the concatenation of the features from the same window with the different filter sizes. The non linear activation function helps in limiting the vector values to specified range. For e.g.: sigmoid function maps the numerical value between the range $[-1.\ 1]$. The results for the various activation functions are shown in Table 11.

5.3.4 Optimization

The gradient descent method is used to optimize the training network, which aims at minimizing the error function that is calculated when the errors are backpropagated to the

Table 9 Effect of filter size

Filter size	Accuracy(SMS Spam)	Accuracy (Twitter)
4	98.83	95.48
5	99.01	94.40

Table 10 Effect of no. filter

No. of Filters	Accuracy(SMS Spam)	Accuracy (Twitter)
32	98.83	94.50
64	98.83	95.48
128	99.01	94.70

previous layer. In the present work, we have used Adagrad method for optimization that adapts learning rate according to the parameters. For frequently occurring parameters larger updates are performed while smaller update done for infrequent parameters. The default learning rate for *Adagrad* is 0.01. The results for various optimization techniques are shown in Table 12.

5.3.5 Dropout rate

The major disadvantage with neural networks is that they tend to overfit especially with the low volume of data. One solution to this problem is to decrease the size of the network rather than increase the amount of data. We optimise the value of the dropout to avoid overfitting in the proposed model. Figure 4 and Table 13 shows the effect of dropout in the proposed model.

5.3.6 Number of features

Features are the words in the corpus that affect the classification judgment. The number of features to be fed in the model need to be limited to the most frequently occurring words rather than taking all the features. This helps to reduce the overfitting due to infrequent words being discarded and they do not participate in the classification process.

In the current architecture, only one convolutional layer with the filter length 4 is used along with a single LSTM layer. The Tables 14 and 15, and Figs. 5 and 6 show the effect of a number of features on the SSCL model.

The number of features performing best in the model remains same in the SMS Spam dataset while it is 8000 in case of Twitter dataset performing even better than the CNN model.

5.3.7 LSTM units

The optimized number of LSTM units working on the word vectors of a text sentence is 100 as can be seen from the evaluation results in Table 16 and Fig. 7. The regularization of the model is performed by adding a dropout layer that drops a number neuron from embedding layer. The dropout of 0.1 and 0.2 gives the best performance for SMS Spam dataset and Twitter dataset respectively as tabulated in Table 14.

Table 11 Effect of activation function

Activation Function	Accuracy(SMS Spam)	Accuracy (Twitter)
tanh	98.74	92.93
ReLU	99.28	95.09
Sigmoid	98.83	92.73

Table 12 Effect of optimization

Optimization	Accuracy(SMS Spam)	Accuracy (Twitter)
Adagrad	99.01	95.09
Adadelta	97.66	88.83
SGD	86.54	83.02
RMSprop	99.01	93.42
Adam	98.20	94.79

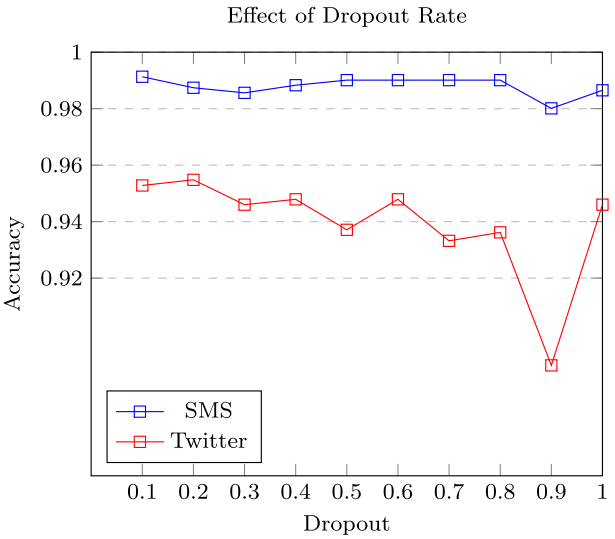

Fig. 4 Effect of dropout rate on accuracy

Table 13 Effect of dropout rate

Dropout	Accuracy(SMS Spam)	Accuracy (Twitter)
0.1	99.01	95.28
0.2	98.74	95.48
0.3	98.56	94.60
0.4	98.83	94.79
0.5	99.01	93.71
0.6	99.01	94.79
0.7	99.01	93.32
0.8	99.01	93.62
0.9	99.01	88.91
1.0	98.65	94.60

Table 14 Effect of number of features

No. of Features	Accuracy (SMS Spam)
5000	99.01
6000	98.69
7000	98.92
8000	98.20

The final set of optimized parameters that are obtained after performing extensive experiments are shown in Table 17. The experiments are performed using these parameters to train *SLSTM* model and the results are compared with the traditional machine learning models as well as deep learning techniques described in CNN and LSTM.

6 Results and discussions

The Tables 18 and 19 describes all the results and performance of the traditional machine learning models along with the proposed approaches on SMS and Twitter dataset respectively. Existing benchmark spam classification methods uses the features extracted from the data by using various feature extraction methods and further use them for classification algorithms. The random forest outperforms other machine learning models with 100 estimators giving the accuracy of 97.85% and 93.43% in SMS and Twitter dataset respectively. However, as the size of the dataset increases which is more likely in case of social networking data, the complexity increases many folds. The simpler yet effective methods like SVM, Naive Bayes give quite well accuracy but the main disadvantage of these method is that they are not able to learn lower level features which is the main advantage of the deep learning methods.

Deep learning methods work on feature representation and constantly learn and fine tune the features which is shown in the Tables 18 and 19. The SCNN and SLSTM both work on word vectors and we have added a semantic meaning to these word vectors by using Word2Vec, WordNet and ConceptNet. The CNN uses the kernel which slides along the features and the weights and parameters are shared. Due to this, it works better than artificial neural network even though it is not a fully connected neural network. LSTM processes the features sequentially learning and storing the information in its memory which can be seen from the experimental results.

The results of the proposed model on both SMS and Twitter datasets are shown in Tables 18 and 19. Experimental results show that the proposed approach performs better

Table 15 Effect of number of features

No. of Features	Accuracy (Twitter)
5000	94.79
8000	95.48
10000	94.21
14000	93.91

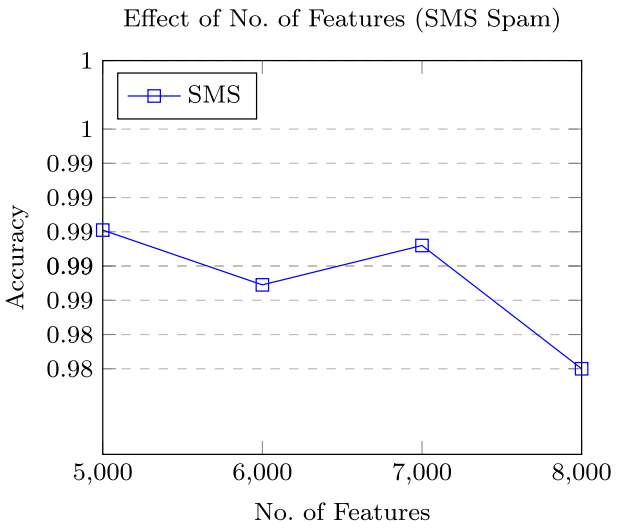

Fig. 5 Effect of number of features in SMS dataset

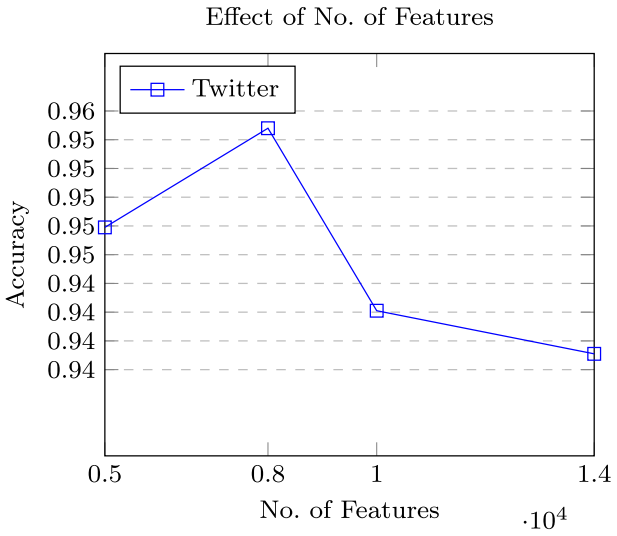

Fig. 6 Effect of no. of features for Twitter Dataset

Table 16 Effect of no. of LSTM units

LSTM Units	Accuracy(SMS Spam)	Accuracy (Twitter)
50	98.65	94.30
100	99.01	95.48
150	98.47	94.21
200	98.83	93.13

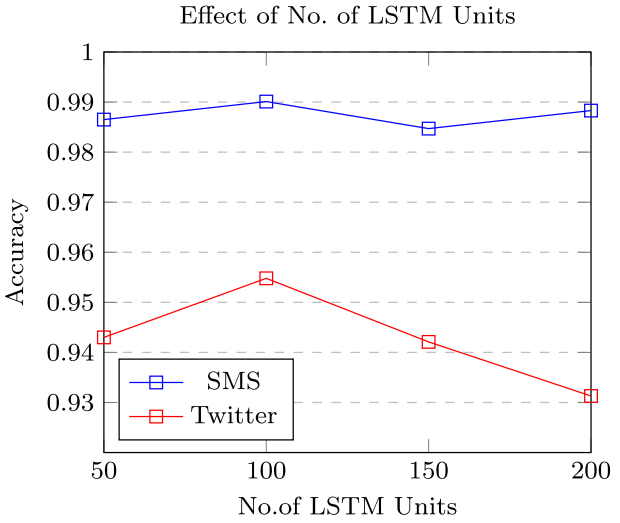

Fig. 7 Effect of no. of LSTM units on accuracy

Table 17 Optimized parameters

Parameter	Value(SMS Spam)	Value (Twitter)
No. of filters	128	54
filter length	5	4
LSTM units	100	100
Dropout	0.1	0.2
No. of Features	5000	8000
Activation Function (CNN)	ReLU	ReLU
Activation Function (LSTM)	Sigmoid	Sigmoid
Epochs	10	10

Table 18 Performance comparison of SSCL and other classifiers: SMS Spam

Classifier	Precision	Recall	Accuracy	F Score
KNN	91.37	90.40	90.40	88.13
NB	97.64	97.67	97.67	97.65
Random Forest	97.88	97.85	97.85	97.77
ANN	97.41	97.40	97.40	97.40
SVM	97.45	97.49	97.49	97.44
CNN	97.67	99.79	97.93	98.82
LSTM	98.56	99.48	98.29	99.02
SCNN	98.78	99.39	98.65	99.07
SLSTM	98.74	99.35	99.01	99.24
SSCL	98.86	99.77	99.01	99.29

Table 19 Performance comparison of SSCL and other classifiers: Twitter

Classifier	Precision	Recall	Accuracy	F Score
KNN	91.61	91.96	91.96	91.38
NB	91.69	92.06	92.06	91.74
Random Forest	93.25	93.43	93.43	93.04
ANN	91.80	91.18	91.18	91.41
SVM	92.91	93.14	93.14	92.97
CNN	92.80	98.93	92.73	95.76
LSTM	93.58	98.22	92.93	95.84
SCNN	94.80	98.62	94.40	96.64
SLSTM	95.54	98.37	95.09	96.84
SSCL	95.88	98.55	95.48	97.13

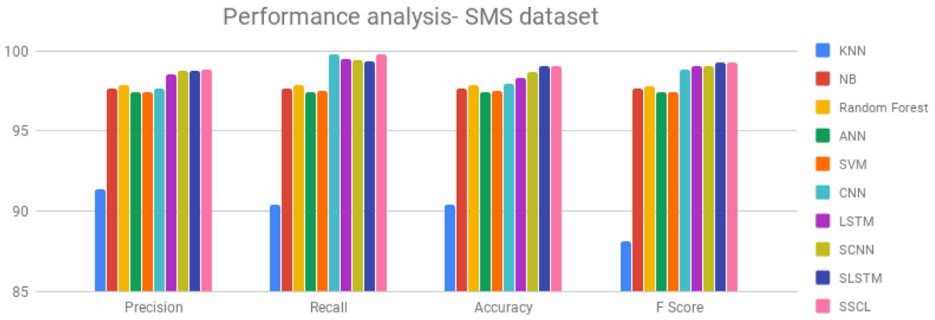

Fig. 8 Performance comparison of proposed approaches with other methods on SMS dataset

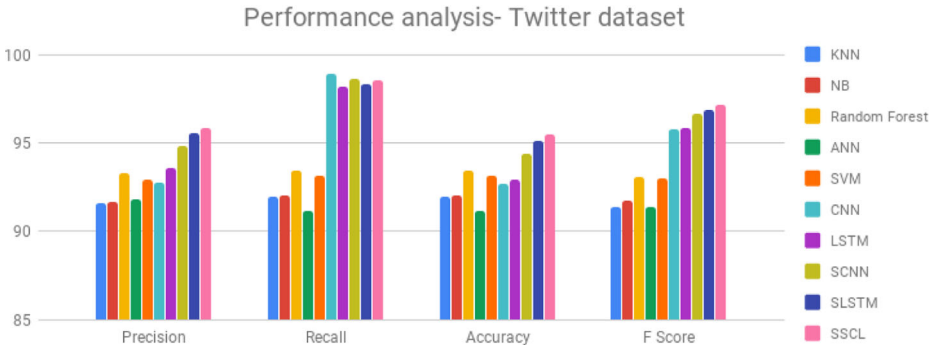

Fig. 9 Performance comparison of proposed approaches with other methods on Twitter dataset

than the traditional machine learning based models. It is also observed from the experimental results that the proposed deep learning architectures performs better than their corresponding models when we do not include the semantic layer within. We also experimented with two other models i.e. *Semantic Convolutional Neural Network (SCNN)* and *Semantic Long-Short Term Memory (SLSTM)*, which are considered by taking each neural network with the semantic embedding layer. This semantic embedding layer is enhancement of word representation using *WordNet* and *ConceptNet* which helps to better initiate the deep architectures since the volume and the length of text data is less. Experimental results show that the proposed approach outperforms these single model also which clearly show that the stacking of CNN and LSTM models improves the performance of the spam detection on both the datasets as shown in Figs. 8 and 9.

Proposed hybrid model produces better results in comparison to the traditional model giving the best accuracy, the accuracy of the SSCL model is increased by 1.16% in the case of SMS spam dataset and 2.05% in the case of Twitter dataset. When comparing the *SSCL* with *SLSTM*, there is a significant increase in performance of the spam detection on the Twitter dataset. In case of SMS Spam dataset, there is an increase in precision, recall and F1 score.

7 Conclusion

Spam detection in social media is very challenging problem mainly due to short-text, use of very noisy and irregular language on the social media. In this paper, we focus on spam detection in social media using deep learning approach. Existing successful approaches mainly focused on long email messages but spam detection in short noisy text such as in Twitter is very challenging due to sparseness problem and irregularity in the language used on social media. We proposed a deep learning based approach consisting of CNN and LSTM neural architectures. We also proposed to use *WordNet* and *ConceptNet* to improve the coverage in Embedding space and provide better initialization for further deep learning architecture. Experimental results show that the proposed approach outperforms all other approaches on two datasets i.e. SMS Dataset and Twitter Dataset.

References

1. Agarwal, B., Mittal, N.: Sentiment analysis using conceptnet ontology and context information. In: Prominent Feature Extraction for Sentiment Analysis. Springer. https://doi.org/10.1007/978-3-319-25343-5 (2016)
2. Almeida, T.A., Yamakami, A., Almeida, J.: Evaluation of approaches for dimensionality reduction applied with naive bayes anti-spam filters. In: International Conference on Machine Learning and Applications, 2009. ICMLA'09, pp. 517–522. IEEE (2009)
3. Androutsopoulos, I., Koutsias, J., Chandrinos, K.V., Spyropoulos, C.D.: An experimental comparison of naive bayesian and keyword-based anti-spam filtering with personal e-mail messages. In: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and development in information retrieval, pp. 160–167. ACM (2000)
4. Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder–decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, pp. 1724–1734 (2014)

5. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. J. Mach. Learn. Res. **12**, 2493–2537 (2011)
6. Cournane, A., Hunt, R.: An analysis of the tools used for the generation and prevention of spam. Comput. Secur. **23**(2), 154–166 (2004)
7. DeBarr, D., Wechsler, H.: Spam detection using clustering, random forests, and active learning. In: Sixth Conference on Email and Anti-Spam. Mountain View (2009)
8. Devlin, J., Kamali, M., Subramanian, K., Prasad, R., Natarajan, P.: Statistical machine translation as a language model for handwriting recognition. In: International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 291–296. IEEE (2012)
9. Gao, Y., Mi, G., Tan, Y.: Variable length concentration based feature construction method for spam detection. In: International Joint Conference on Neural Networks (IJCNN), pp. 1–7. IEEE (2015)
10. Grier, C., Thomas, K., Paxson, V.: Zhang, M.: spam: the underground on 140 characters or less. In: Proceedings of the 17th ACM Conference on Computer and Communications Security, pp. 27–37. ACM (2010)
11. Havasi, C., Speer, R., Alonso, J.: Conceptnet 3: a flexible, multilingual semantic network for common sense knowledge. In: Recent Advances in Natural Language Processing (RANLP). John Benjamins Philadelphia, pp. 27–29 (2007)
12. Healy, M., Delany, S.J., Zamolotskikh, A.: An assessment of case base reasoning for short text message classification. In: Proceedings of the 15th Irish Conference on Artificial Intelligence and Cognitive Sciences (AICS'04), pp. 9–18 (2004)
13. Jain, G., Sharma, M.: Social media: a review. In: Information Systems Design and Intelligent Applications, pp. 387–395. Springer (2016)
14. Jain, G., Sharma, M., Agarwal, B.: Optimizing semantic lstm for spam detection. Int. J. Inf. Technol. 1–12 (2018)
15. Jain, G., Sharma, M., Agarwal, B.: Spam detection on social media using semantic convolutional neural network. Int. J. Knowl. Disc. Bioinfo **8**(1), 12–26 (2018)
16. Karami, A., Zhou, L.: Improving static sms spam detection by using new content-based features. In: Twentieth Americas Conference on Information Systems, Savannah, pp. 1–9 (2014)
17. Kim, C., Hwang, K.B.: Naive bayes classifier learning with feature selection for spam detection in social bookmarking. In: ECML PKDD Discovery Challenge, p. 32 (2008)
18. Kim, J., Chung, K., Choi, K.: Spam filtering with dynamically updated url statistics. IEEE Secur. Priv. 5(4) (2007)
19. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1746–1751. Association for Computational Linguistics (2014)
20. Kolari, P., Finin, T., Joshi, A.: SVMS for the blogosphere: Blog identification and splog detection. In: AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs, pp. 92–99 (2006)
21. Lai, S., Xu, L., Liu, K., Zhao, J.: Recurrent convolutional neural networks for text classification. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI), vol. 333, pp. 2267–2273 (2015)
22. Lei, T., Barzilay, R., Jaakkola, T.: Molding cnns for text: non-linear, non-consecutive convolutions. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1565–1575. Association for Computational Linguistics (2015)
23. Levine, J.R.: Experiences with greylisting. In: Second Conference on Email and Anti-Spam (CEAS), pp. 1–2 (2005)
24. Ma, J., Gao, W., Mitra, P., Kwon, S., Jansen, B.J., Wong, K.F., Cha, M.: Detecting rumors from microblogs with recurrent neural networks. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI), pp. 3818–3824 (2016)
25. Martinez-Romo, J., Araujo, L.: Detecting malicious tweets in trending topics using a statistical analysis of language. Expert Syst Appl **40**(8), 2992–3000 (2013)
26. Mccord, M., Chuah, M.: Spam detection on twitter using traditional classifiers. In: International Conference on Autonomic and Trusted Computing, pp. 175–186. Springer, Berlin (2011)
27. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv:1301.3781 (2013)
28. Miller, G.A.: Wordnet: a lexical database for english. Commun. ACM **38**(11), 39–41 (1995)
29. Miller, Z., Dickinson, B., Deitrick, W., Hu, W., Wang, A.H.: Twitter spammer detection using data stream clustering. Inf. Sci. **260**, 64–73 (2014)
30. Mou, L., Peng, H., Li, G., Xu, Y., Zhang, L., Jin, Z.: Discriminative neural sentence modeling by tree-based convolution. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 2315–2325. Association for Computational Linguistics (2015)
31. Sabri, A.T., Mohammads, A.H., Al-Shargabi, B., Hamdeh, M.A.: Developing new continuous learning approach for spam detection using artificial neural network. Eur. J. Sci. Res. **42**(3), 525–535 (2010)

32. Sainath, T.N., Vinyals, O., Senior, A., Sak, H.: Convolutional, long short-term memory, fully connected deep neural networks. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4580–4584. IEEE (2015)
33. Silva, R.M., Almeida, T.A., Yamakami, A.: Artificial neural networks for content-based web spam detection. In: Proceedings on the International Conference on Artificial Intelligence (ICAI), p. 1 (2012)
34. Socher, R., Bauer, J., Manning, C.D., Manning, C.D., Andrew, Y.N.: Parsing with compositional vector grammars. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pp. 455–465 (2013)
35. Stern, H.: A survey of modern spam tools. In: The Fifth Conference on Email and Anti-Spam (CEAS), pp. 1–10 (2008)
36. Stringhini, G., Kruegel, C., Vigna, G.: Detecting spammers on social networks. In: Proceedings of the 26th Annual Computer Security Applications Conference (ACSAC'10), pp. 1–9. ACM (2010)
37. Tai, K.S., Socher, R., Manning, C.D.: Improved semantic representations from tree-structured long short-term memory networks. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, pp. 1556–1566. Association for Computational Linguistics (2015)
38. Thomas, K., Grier, C., Ma, J., Paxson, V., Song, D.: Design and evaluation of a real-time url spam filtering service. In: 2011 IEEE Symposium on Security and Privacy (SP), pp. 447–462. IEEE (2011)
39. Tseng, C.Y., Chen, M.S.: Incremental SVM model for spam detection on dynamic email social networks. In: International Conference on Computational Science and Engineering, (CSE'09), vol. 4, pp. 128–135. IEEE (2009)
40. Wang, H.B., Yu, Y., Liu, Z.: SVM classifier incorporating feature selection using ga for spam detection. In: Embedded and Ubiquitous Computing–EUC, vol. 2005, pp. 1147–1154 (2005)
41. Wu, F., Shu, J., Huang, Y., Yuan, Z.: Co-detecting social spammers and spam messages in microblogging via exploiting social contexts. Neurocomputing **201**, 51–65 (2016)
42. Wu, T., Liu, S., Zhang, J., Xiang, Y.: Twitter spam detection based on deep learning. In: Proceedings of the Australasian Computer Science Week Multiconference, ACSW '17, pp. 3:1–3:8. ACM, New York (2017)
43. Zhang, L., Zhu, J., Yao, T.: An evaluation of statistical spam filtering techniques. ACM Trans. Asian Lang. Inf. Process. (TALIP) **3**(4), 243–269 (2004)
44. Zhang, Y., Wang, S., Phillips, P., Ji, G.: Binary pso with mutation operator for feature selection using decision tree applied to spam detection. Knowl.-Based Syst. **64**, 22–31 (2014)