Computer Science and Information Technologies
Vol. 1, No. 2, July 2020, pp. 84~92
ISSN: 2722-3221, DOI: 10.11591/csit.v1i2.p84-92

Insult detection using a partitional CNN-LSTM model

Mohamed Maher Ben Ismail
Computer Science Department, College of Computer and Information Sciences, King Saud University, Riyadh, Kingdom of Saudi Arabia

Article Info

Article history:

Received Sep 23, 2019
Revised Apr 20, 2020
Accepted May 9, 2020

Keywords:

Deep learning,
Insult detection,
Social networks,
Supervised learning

ABSTRACT

Recently, deep learning has been coupled with noticeable advances in Natural Language Processing related research. In this work, we propose a general framework to detect verbal offense in social networks comments. We introduce a partitional CNN-LSTM architecture in order to automatically recognize verbal offense patterns in social network comments. Specifically, we use a partitional CNN along with a LSTM model to map the social network comments into two predefined classes. In particular, rather than considering a whole document/comments as input as performed using typical CNN, we partition the comments into parts in order to capture and weight the locally relevant information in each partition. The resulting local information is then sequentially exploited across partitions using LSTM for verbal offense detection. The combination of the partitional CNN and LSTM yields the integration of the local within comments information and the long distance correlation across comments. The proposed approach was assessed using real dataset, and the obtained results proved that our solution outperforms existing relevant solutions.

Corresponding Author:

Mohamed Maher Ben Ismail,
Computer Science Department,
College of Computer and Information Sciences,
King Saud University, Riyadh, KSA.
Email: ts@ee.uad.ac.id

1. INTRODUCTION

The growth of the world population as well as the technological advances have led a new era of communication and socialization through virtual platforms such as YouTube, Instagram, Twitter, and LinkedIn. Nowadays, billions of people all around the world joined social networks which require a basic knowledge of the computer fundamentals. Besides, the outspread use of the smart devices along with the excessive use of social networks has granted them the ability to form various virtual societies where people can continuously exchange ideas, interests and concerns. This resulted in a new lifestyle where they regularly follow, share and get updates on events that are held in their actual society. In fact, people are using social networks for various purposes regardless of their ethnicity, nationality, education and background. In particular, social networks enable the users to interact with their peers. Such involvement of the worldwide population in digital society has yielded various challenges and side effects. For instance, security, spam detection and privacy protection has emerged as critical challenges facing social network professionals and companies. Governments, such as in Saudi Arabia, have established the Communications and Information Technology Commission to overcome these challenges and to cope with the radical changes that rapidly happen in the digital world [1]. they have also regulated Anti-Cyber Crime Law to be implemented through government department such as Ministry of Interior [2] and Public Prosecution [3] to avoid any unethical misuse of the social networks and to prevent any violations that may occur within the cyberspace. This proves that some

of the challenges faced by the digital communities are critical and require real efforts to limit their impact on people daily life. Despite these efforts, some social media users break the communication ethics code while messaging, discussing or commenting on social media. This behavior can be attributed to many factors such as their psychological condition, low education level or living environment. In particular, textual insult is a typical illustration of this problem. A typical textual insult consists in the use of vocabulary which harms the user being communicating with. Such offense is often hard to sense because its patterns exhibit high variability. Typically, it can be direct insult, intimidation, shout or threat. However, whatever the form it takes, it remains unacceptable for the majority of users. Moreover, conservative societies are more sensitive to such phenomena. Thus, aggressive behavior through threats by implying abuse such as "Don't you dare do that or I'll punch your lights out!" is also not accepted. Similarly, fowl name usage such as "You're a stupid good for nothing!" is not tolerated. Despite authority efforts to suite users who offend others in social media through appropriate legislation, increasing amount of insults are regularly reported on social networks. Thus, verbal offenses have become the issue that most of the users face when connecting to virtual societies. Sadly, users must handle manually such concern. For example, the administrators of Facebook pages should screen all comments on every single post and discard insults. This manual solution is subjective and labor demanding especially when the number of comments to handle is considerably large. Moreover, given the continuously gpowing number of users, blocking the user along with reporting them to the moderators has also become an obsolete alternative. Therefore, solutions able to automatically detect verbal insult emerged as an urgent need. One of the earliest efforts to solve this problem in an unsupervised manner was to specify a list of prohibited words so that if any words of the list appeared in the user message, the message or comment will be rejected. Typically, such solution rely on a static dictionary along with some socio-linguistic patterns and semantic rules [4-6]. However its main drawback consists in its inability to decide intelligently if the text is an insult or not. For example, if we consider the following two comments: "This idea stupid" and "You are stupid". The second one is an insult while the first comment is not. A typical prohibited list based method cannot discriminate between them, and would either reject or tolerate both comments. Another alternative to detect verbal abuse consists in the formulation of the problem as text mining and supervised learning problem (classification). In fact, thes classifiers are intended to determine whether a comment is an insult or not. Commonly, some training comments are first used to learn the mapping between the annotated comments and the two predefined classes. Then, the resulting mapping model is used to automatically predict the class value of the unlabeled comments. Despite researcher's effort to solve various real world applications using supervised learning algorithms [7-11], a limited number of solutions able to detect insults in social network comments in an unsupervised manner has been outlined so far. Lately, deep learning have proved to be promisingly accurate in predicting classes in various applications. In fact, various deep learning models have been introduced and deployed to overcome text classification challenges. In particular, the Recurrent Neural Network (RNN) was designed to capture semantic information sequentially through fixed length hidden layer vectors which process consecutive time-step words [12]. However, such model may exhibit bias towards later words when encoding the overall sentence/comment semantics. This RNN drawback can interpreted as a result of an exploding gradient which yields large updates of the model weights. To address this issue, the Long Short-Term Memory (LSTM) network was introduced in [13] to better capture the short and long time dependencies. Moreover, it was intended to address the gradient explosion and gradient diffusion problems inherited from typical RNNs [13]. In this paper we propose a partitional CNN-LSTM architecture to build a supervised learning model able to detect if a given comment/sentence represents a verbal offense. The proposed local CNN processes the user comment as a subsequence rather than handling the whole comment/sentence as done using the typical CNN models. In particular, it partitions the input comments into sequences in a way that the relevant information in each partition is captured and weighted based on its relevance to the offense. The captured local information is then sequentially exploited using LSTM and coupled with the global dependency extracted using the typical CNN in order better model verbal offense semantic.

2. RELATED WORKS

Insult detection in social network comments is intended to reject comments conveying insulting messages in an automatic manner. It was introduced as an alternative to support and/or substitute the manual effort of the virtual community administrators. Typically, supervised machine learning techniques have been adopted by the recent verbal offense detection approaches. In the following sub-sections, we outline the state of the-art text classification approaches based on supervised learning techniques as well as the relevant deep learning techniques, respectively.

2.1. Verbal offense detection using supervised learning techniques

Recently, many researchers have contributed to introduce various solutions to address the problem of automatic verbal offense detection for social network comments. The authors in in [12] presented a solution that adopts a static socio-linguistic based dictionary to detect the comments including words from the dictionary. One should note that the reported results showed low coverage and high false positive rates. In [5], the authors outlined a discrimination approach between regular and insult statements based on sentences parsing and semantic rules usage. The solution introduced in [6] to reject insulting comments is based on the bag-of words features along with a dictionary that includes the abusing language. In [14], a linguistic analysis based insult detection solution for Thai textual conversations was proposed. The authors in [15] proposed an online detection system that detects harassment. The main goal was to determine whether a comment represents an harassment or not. Note that they formulated the harassment detection as a sentiment analysis problem. In [16], the outlined system aims to categorize the user comments as bullying or not using a Multi-Criteria Evaluation System (MCES) which revolves around the concept of weighting words based on a score or a numerical value. In [17], the researchers introduced a solution that relies on the linguistic regularities captured in profane language using statistical topic modeling. A stochastic gradient descent classifier was used in [18] to detect insults in usergenerated Arabic newspaper commentary. The solution was able to detect modern standard Arabic and colloquial Egyptian Arabic. In [19], the authors proposed a system that relies on multi-level classification to detect flame in an automatic manner. This research applied machine learning techniques for automatic offensive language detection. the authors used supervised learning methods, namely the Naive Bayes and the Support Vector Machine (SVM) to assign comments to on the “sexual” or “racist” category. As one can notice the state-of-the-art insult detection approaches above typically use supervised learning algorithms to automatically map the social media comments to the predefined classes. Since such verbal offense detection solutions are relatively rare, we additionally cover relevant text classification approaches.

2.2. Typical text classification

Typical text classification systems rely on text representation and feature selection for a better discrimination between the predefined text categories. Besides, the feature selection/reduction can also be conducted to reduce the feature space dimensionality. In particular, the Latent Dirichlet Allocation [20] has been exploited to determine the corpus topics, and define the feature space accordingly. However, this approach is constrained by the large size of the resulting vocabulary compared to the standard Bag Of Words (BOW) representation. In fact, despite the promising performance achieved in text mining applications using word embedding, the tradition Bag of Words (BoW) model is still adopted in various applications and proved to perform relatively well. The BoW model encodes only the keywords occurrence frequency in a given set of documents. In particular, TF-IDF representation proved to be successful in capturing the patterns among the text semantic categories. Note that no information on the structure of words in a given document is enclosed in such representation [21]. In other words, sparse representations remains challenging from the computational and learning point of views. A simple alternative to limit the effect of the data sparsity consists in discarding the keywords with sparsity higher than 99% which reduces simultaneously the data dimensionality. Other researchers used graph representation for text data and coupled it with appropriate distance/similarity measures [22] in order to use graph mining algorithms. Specifically, the latter algorithms were intended to mine frequent sub-graphs in the document collection to construct the feature space [21]. However, such representation usually exhibits high computational and space costs. On the other hand, hierarchical classification has been also adopted for text classification [20], [23]. In [7], a review on the use of supervised learning for opinion mining during the last decade was done. The researchers in [24] introduced an emotion detection system that is intended to recognize nastiness and sarcasm in online conversation. Besides, the authors investigated the use of different feature sets along with two supervised learning algorithms to improve the overall classification performance. The work in [25] introduced the keystones of an irony detection approach which takes into consideration the customer feedback in the learning of the classification model. In [9-11], the authors outlined the state-of-the-art solutions proposed to recognize regular emails and detect junk ones [8]. Despite such considerable efforts to overcome real applications challenges, it can be admitted that there is no universal solution for all classification challenges. In other words, it makes no sens to claim that a classification technique overtakes the others in all applications [4]. Therefore, deep learning based classification emerged as a promising alternative to address the text classification problems.

2.3. Text classification based on CNN and LSTM

Given their ability to learn the statistical properties of the images, CNN have been widely used in image categorization applications [26]. Specifically, CNNs’ convolution operator captures the lowly variant dependency between neighboring pixels in the image regions. Such statistical image characteristics can be also

found in textual comments since neighboring words in a given comment exhibit some dependency. Therefore, the keywords included in a comment should be encoded in order to be equivalent to the image pixels and fed to the CNN [27]. Typical text representation techniques are used to index the collection of keywords that are used in the textual comments. Then, the resulting matrix is transformed into a lower dimensional representation after going through the embedding layer [28]. Such keyword representation can be obtained by deploying a distribution over the keyword which results in a fixed length dense vector. This 'randomized' approach is tuned through the CNN training phase. One should note that, dense keywords vectors of fixed length obtained using keyword embedding methods like GloVe [22] and word2vec [29] can also be adopted. Typically, keyword embedding requires a training phase using large collections. For instance, the training of the word2vec model relies on a collection of 100 billion words which yielded a 3 million keyword vocabulary. Various semantic composition approaches have been introduced to better represent the documents/comments in text classification applications. In particular, deep learning paradigms, such as RNN, CNN and LSTM, have been adopted to design robust neural networks. In [30], a typical CNN network which comprises one convolution layer including filters of various width. In addition, a max pooling and fully connected layers are associated for sentiment classification. Other researchers associated the autoencoder with RNN to learn a meaningful representation in the context of statistical machine translation [31]. The authors in [32] used matrices to handle the nodes of the tree structure of their RNN. This yielded better representation of the sentiment expressed in the considered sentences. Lately, as outlined in [33], cell blocks of LSTM model were integrated in RNN network to represent the non leaf nodes of the network tree structure. The resulting model was intended to better capture the semantic meaning of the text sentences. In [34], the authors proposed a BoW based CNN that relies on a a convolution layer and feed it the bag-of-word features. In addition, they introduced a Sequential CNN that is intended to encode the keywords sequential information through the concatenation of a single vector of multiple keywords. The researchers in [35] outlined a document representation approach based on neural networks that can learn the relationships between sentences. Specifically, their approach couples CNN and LSTM with word embedding to represent the sentences. Besides, they adopted the Gated Recurrent Unit (GRU), which is an extension of LSTM, to capture the sentence's semantics for a more accurate document categorization. Another deep memory network was used in [36] to model the user meta-data. Specifically, a LSTM was used for the document representation, while the deep memory network was deployed to automatically rate new documents. In [37], the authors introduced an attention-based LSTM network for a document level based sentiment prediction. Note that resulting solution supports the English and Chinese languages. In [38], the researchers depicted various variations of the CNN based sentiment classification approach. Particularly, they investigated the CNN-static where they pretrain and fix the word embedding apriori, the CNN-rand where they randomly initialize the word embedding, and the CNN-multichannel where they used several word embedding sets. The authors in [39] designed a regional CNN-LSTM architecture that is intended to map the learned text features into a set of predefined ratings categories. Similarly in [40], a CNN and LSTM based deep neural network was constructed and associated with linguistic embedding and word2vec to classify sentences as "feeling" or "factual". In [41], the researchers outlined a neural network architecture based on two CNNs where two hidden layers used for the feature representation and fed with both the annotated and unannotated instances. The resulting model was intended to generalize the sentence embedding for an accurate sentiment classification. In order to recognize the sentence sentiment accurately, the authors in [42] presented a model that exploits the linguistic resources and takes into consideration information such as the negation words, sentiment lexicon, and intensity words into the LSTM network.

3. PARTITIONAL CNN-LSTM MODEL

The proposed local CNN-LSTM architecture is depicted in Figure 1. Note that to classify textual comments using convolutions, we converted the text instances into images. Therefore, the word2vec that consists in a two-layer neural net was first used to process the comment collection. More specifically, the comments were converted into sequences of keywords vectors of length d using word embedding [41]. The resulting numerical vectors are then fed into the the deep neural network. In particular, the proposed local CNN model splits a comment into *M* partitions$\{p_1, p_2, \dots, p_M\}$. Relevant features are then extracted from these partitions. Specifically, the convolutional and max pooling layers process sequentially the input vectors in order to learn the relevant features. Finally, the LSTM is used to incorporate sequentially the obtained local features across the partitions to form the overall comment vector to be automatically categorized as insult or not. The convolutional layer is initially intended to form the local ngram features for each partition. Let the partition matrix be $S \in R^{d \times M}$ where M is the sequence vocabulary size, and d is the keyword vectors dimensionality. As illustrated in Figure 1, the keyword vectors in the partitions $p_i = \{w_1^{p_i}, w_2^{p_i}, \dots, w_I^{p_i}\}$, $p_j = \{w_1^{p_j}, w_2^{p_j}, \dots, w_J^{p_j}\}$ and $p_k = \{w_1^{p_k}, w_2^{p_k}, \dots, w_K^{p_k}\}$ are aggregated to get the partition matrices x^{p_i}, x^{p_j} and x^{p_k}.

As one can notice, *C* convolutional filters are used for each partition to extract the local *n*-gram features. In a sequence of *K* keywords $x_{n:n+K-1}$, the deployment of a filter $H_{t,l \le t \le T}$ results in a feature map y_n^t:

$$y_n^t = \mathrm{f}(W^t o x_{n:n+K-1} + b^t) \quad (1)$$

Where the operator *o* represents a convolution, *b* and $W \in R^{K \times l}$ are the bias and the weight matrices respectively. On the other hand, l is the dimension of the keyword vector, ω is the filter length and *f* denotes the ReLU function. The feature maps $y^t = y_1^t, y_2^t, \dots, y_{N-K+1}^t$ of the filter H_t are obtained after a filter scans progressively from $x_{1:K-1}$ to $x_{N+K-1:N}$. Note that the comment partitions exhibit variable text lengths which yields variable dimensions for y^t . Next to the input layer of length *N*, the output of the convolutional layer is subsampled in the Max-pooling layer. In particular, pooling is performed through the application of a max function to the output of each filter. This operation is intended to reduce the computational cost of the upper layers and discard the non-maximal values. In addition, it processes the different partitions and captures the local dependency to determine the most salient information. The resulting partition vectors are then provided to a sequential layer. For this sequential layer, the inter-partition long-distance dependency is captured by a sequential integration of the partition vectors into the comments vectors. Note that the LSTM is introduced in this layer in order to address the typical RNN gradient vanishing or exploding problem. Once all partitions are sequentially traversed by the LSTM memory cell, the last sequential layer hidden state can be perceived as the comment representation for insult detection. Finally, a typical Softmax classifier is adopted for the last layer. The minimization of the mean squared error between the ground truth class values and the predicted is used to train the local CNN-LSTM. Let $X = x^1, x^2, \dots, x^m$ be a training set of text matrix, and $y = y^1, y^2, \dots, y^m$ be the corresponding class values. On the other hand, we define the loss function as:

$$L(X, y) = \frac{1}{2m} \sum_{i=1}^{m} \| h(x_i) - y_i \|^2 \quad (2)$$

Besides, the back propagation algorithm in [43] based on the stochastic gradient descent (SGD) is used in the training phase in order to optimize the network parameters.

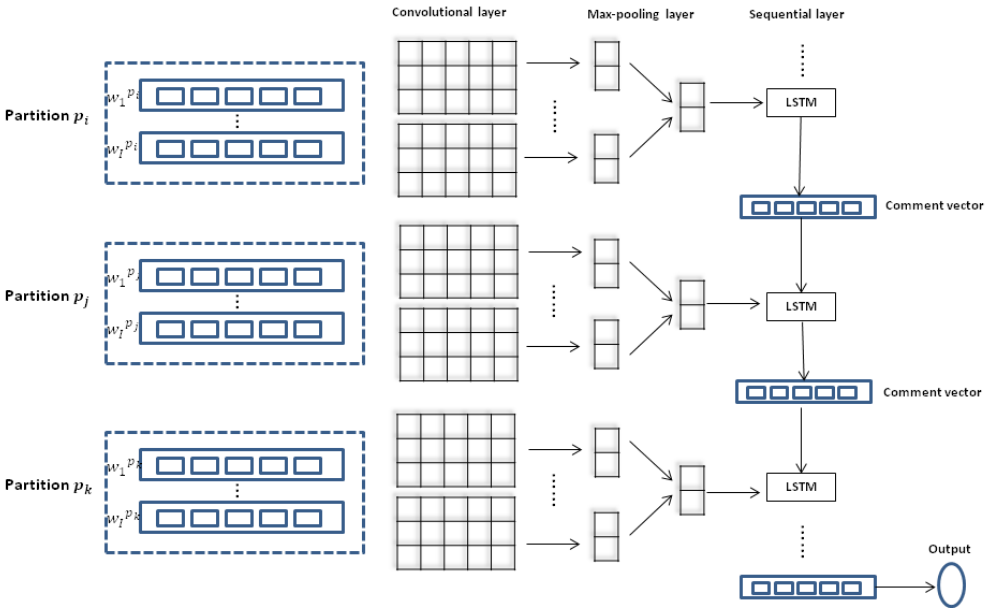

Figure 1. Architecture of the proposed model

4. EXPERIMENTS

We conducted a range of experiments to evaluate the performance of the proposed approach. Particularly, we used KAGGLE dataset [44] which represents a collection of comments from various social media. The 6183 comments which compose this dataset belong to the "insult" and "insult-free" categories. First, these comments were pre-processed in order to discard some encoding parts that may affect the results.

Specifically, the comments were tokenized and converted to lowercase. In addition, all punctuation characters were erased. This results in a vocabulary of 15322 keywords. To implement the proposed approach, a network with 1-D convolutional filters of varying widths were trained. Note that each filter width corresponds to the number of keywords the filter can process which corresponds to the n-gram length. In our experiments, we used the pre-trained word embedding model (FastText) [45]. FastText is an English 16 Billion Token Word Embedding support package. This model was adopted to initialize the weights of the embedding layer. This is intended to to build 300-dimension word vectors for all comments. The hyper-parameters of the proposed architecture were optimized based on the performance of the training and validation phases using the search function introduced in [46]. This tuning strategy aims to investigate all candidate parameter combinations, assess the corresponding models and determine the optimal settings. For the considered dataset, the optimal parameters of the proposed network are shown in the Table below:

Table 1. The hyper-parameters of the proposed network architecture

# filters (m)	Filter length (l)	Pool length (n)	Dropout rate (p)	LSTM layer count (c)	LSTM hidden layer (d)	Training batch size (b)/Epochs(s)
64	3	2	0.1	2	200	100/10

In order to assess the performance of the proposed approach, we used the following standard performance measures in all our experiments. Namely, the accuracy was obtained using:

$$Accuracy=(\#CorrectPredictions) /(TotalNumberOfPredictions) \quad (3)$$

As one can see in Figure 2, the validation accuracy attained by the proposed approach is 80.89% with a learning rate of 0.01. On the other hand, the training accuracy reaches 100%.

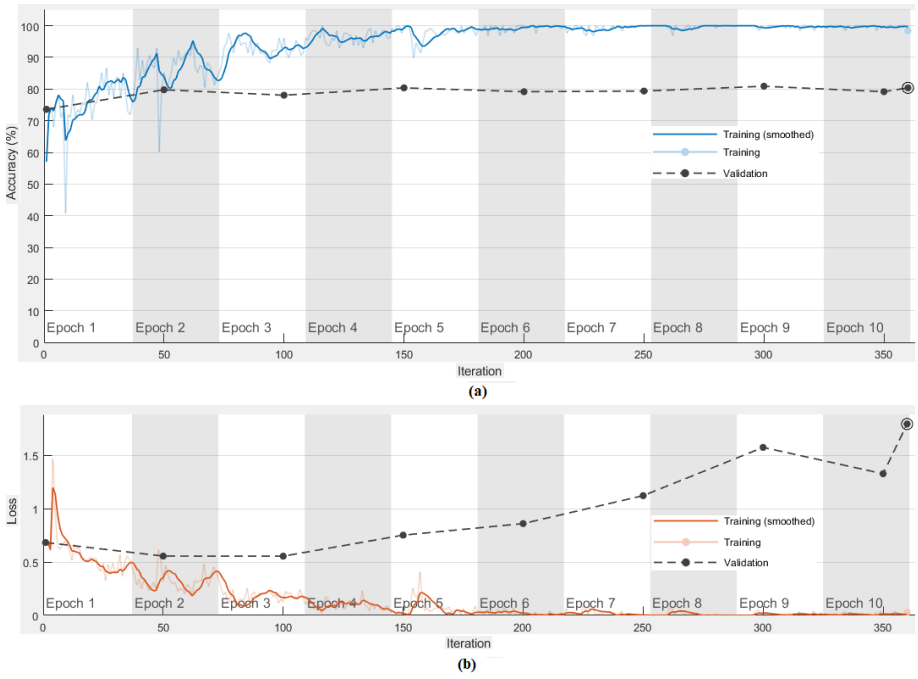

Figure 2. Training progress: (a) accuracy vs iteration. (b) loss vs iteration

Similarly, the Recall and Precision metrics were calculated using:

$$Recall=(\#CorrectlyDetected(Insult)) /(TotalNumberOfInsult) \quad (4)$$

$$Precision=(\#CorrectlyDetected(Insult\text{-}free)) /(TotalNumberOfInsult\text{-}free) \quad (5)$$

In addition, the F-measure (F1 score) was considered and computed using:

$$F1=2 \times (Precision \times Recall) /(Precision+Recall) \quad (6)$$

Table II reports the performance measure attainment achieved using the proposed approach as well as relevant state of the art methods. As it can be seen, the proposed method overtakes the other approaches in terms of Specificity, Accuracy and Precision. In particular, the proposed method based on CNN and LSTM detected about 37% more insult comments than typical CNN-based classification. Note that the CNN-based results were obtained after converting the comment collection into images. Besides, the instances were padded in order to have a constant length. Furthermore, the documents were converted into sequences of keyword vectors using the wor2vec word embedding [29]. Particularly, the implemented network relies on 1-D convolutional filters of varying widths. In other words, the width of each filter fits the n-gram length. In fact, the different branches of convolutional layers of the network handle the multiple n-gram lengths. The CNN network architecture can be summarized as follows:

- Blocks of layers which consist of a convolutional layer, a batch normalization layer, a ReLU layer, a dropout layer, and a max pooling layer were designed to handle the n-gram lengths 2, 3, 4, and 5.
- 200 convolutional filters along with pooling regions were used for each block.
- The input layer was connected to each block.
- The outputs of the blocks were aggregated using a depth concatenation layer.
- A fully connected layer, a softmax layer, and a classification layer were included for the classification task.

Table 1. Performance measures obtained using the method in [47], typical SVM classification, a CNN-based method and the proposed method, respectively

Method	Accuracy	Recall	Precision	F1-measure
Method in [47]	0.598	0.597	0.685	0.638
SVM-based method	0.606	0.223	0.741	0.343
CNN-based method	0.728	0.689	0.742	0.715
Proposed Method	0.834	0.944	0.793	0.862

Furthermore, we conducted a statistical Student t-test [48] using a confidence level of 95%. This test was intended to decide if the means of two decision sets obtained using two different models are reliably different. Thus, if the difference between the mean of the performance measures is statistically significant, then the null hypothesis that assumes that the two samples follow similar distributions is rejected. Specifically, for the *p*-values [49] below 0.05, the classification results were statistically significant. Therefore, the null hypotheses were rejected by the *t*-test as shown in Table 3.

Table 2. T-test results based on the performance measures of the different approaches

	Proposed Method Vs Method in [59]	Proposed Method Vs SVM-based method	Proposed Method Vs CNN-based method
Accuracy	1	1	1
Recall	1	1	1
Precision	1	1	1

Further investigation showed our approach categorizes less accurately non-offensive comments which yields lower sensitivity. Despite this contrast between the specificity and the sensitivity attainment, these results can be considered promising. In fact, for such insult automatic detection problem, one can assume that the True Positive predictions are not as important as the True Negative instances. Specifically, the misclassification of an insulting comment is not considered as critical as the misclassification of a regular one. In addition, the accuracy cannot be a reliable performance measure for this application because the testing data includes 720 verbally offensive comments only out of the 2674 comments,

5. CONCLUSION

In this paper we have proposed a novel approach of automatic insult detection in social network comments. Specifically, we proposed a partitional CNN-LSTM model intended to automatically recognize verbal offense in social network comments. In particular, we designed a partitional CNN and LSTM architecture to map social network comments into "insult" or "regular" categories. In fact, instead of considering a whole document/comments as input as for typical CNN, we partition the comments into parts in order to capture and weight the locally relevant information in each partition. The obtained local information is then sequentially exploited across partitions using LSTM for verbal offense detection. The association of such partitional CNN and LSTM allows the integration of the local within comments information and the long distance correlation across comments. The obtained experimental results proved that the proposed approach overtakes existing relevant approaches.

ACKNOWLEDGMENT

This project was supported by the Research Groups Program (Research Group number RG-1439-033), Deanship of Scientific Research, King Saud University, Riyadh, Saudi Arabia.

REFERENCES

[1] "Communications and Information Technology Commission." [Online]. Available: http://www.citc.gov.sa/ar/Pages/default.aspx [Accessed 1 November 2017].
[2] Xiang, Guang & Fan, Bin & Wang, Ling & Hong, Jason & Rose, Carolyn. "Detecting offensive tweets via topical feature discovery over a large scale twitter corpus." Proceedings of the *21st ACM Conference on Information and Knowledge Management*, Sheraton, Maui Hawaii, pp. 1980-1984, 2012.
[3] L. K. Hansen and P. "Salamon. Neural network ensembles." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 10, pp. 993–1001, 1990.
[4] Ellen Spertus. "Smokey: Automatic recognition of hostile messages." In *Proceedings of the Ninth Conference on Innovative Applications of Artificial Intelligence*, pp. 1058–1065, 1997.
[5] Altaf Mahmud, Kazi Zubair Ahmed, and Mumit Khan. "Detecting flames and insults in text." In *Proceedings of the Sixth International Conference on Natural Language Processing*, 2008.
[6] Amir H. Razavi, Diana Inkpen, Sasha Uritsky, and Stan Matwin. "Offensive language detection using multi-level classification." In *Proceedings of the 23rd Canadian Conference on Artificial Intelligence*, pp. 16–27, 2010.
[7] "Ministry of Interior in Saudi Arabia." [Online]. Available: https://www.moi.gov.sa / [Accessed 1 November 2017].
[8] "Public Prosecution in Saudi Arabia." [Online]. Available: https://www.bip.gov.sa/ [Accessed 1 November 2017].
[9] D. Lewis, K. Knowles. "Threading electronic mail: A preliminary study." *Information Processing and Management*, vol. 33, no. 2, pp. 209–217, 1997.
[10] W. Cohen. "Learning rules that classify e-mail." *AAAI Conference*, 1996.
[11] V. R. de Carvalho, W. Cohen. "On the collective classification of email speech acts." *ACM SIGIR Conference*, 2005.
[12] K. Woods, J. Kegelmeyer, W. P., and K. Bowyer. "Combination of multiple classifiers using local accuracy estimates." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 4, pp. 405–410, 1997.
[13] Y. Bi, D. Bell, H. Wang, G. Guo, and J. Guan. "Combining multiple classifiers using dempster's rule for text categorization." *Applied Artificial Intelligence*, vol. 21, no. 3, pp. 211–239, 2007.
[14] J. L. Elman. "Finding structure in time." *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.
[15] S. Hochreiter and J. Schmidhuber. "Long short-term memory." *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[16] M. Dadvar, D. Trieschnigg, F. Jong, "Experts and Machines against Bullies: A Hybrid Approach to Detect Cyberbullies." *Advances in Artificial Intelligence*, pp 275-281, 2014.
[17] R. Justo, T. Corcoran, S. Lukin, M. Walker, M. Ines Torres. "Extracting Relevant Knowledge for the Detection of Sarcasm and Nastiness in the Social Web." *Knowledge-Based Systems*, 2014.
[18] Guang Xiang Bin Fan Ling Wang Jason I. Hong Carolyn P. Rose., "Detecting Offensive Tweets via Topical Feature Discovery over a Large Scale Twitter Corpus." Proceeding of the *21st ACM international conference on Information and knowledge management (CIKM '12)*, pp. 1980 1984, 2012.
[19] Razavi, Amir, Inkpen, Diana, Uritsky, Sasha & Matwin, Stan. "Offensive Language Detection Using Multi-level Classification." pp. 16-27, 2010.
[20] D. Yin, Z. Que, L. Hong. "Detection of Harassment on Web 2.0." *CAW 2.0*, Spain, 2009.
[21] Wei Wang, Diep Bich Do, and Xuemin Lin. "Term graph model for text classification." In *International Conference on Advanced Data Mining and Applications*. Springer, pp. 19–30, 2005.
[22] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. "Latent Dirichlet Allocation." *Journal of Machine Learning Research*, pp. 993–1022, 2003.
[23] Anand Rajaraman and Jeffrey David Ullman. "Mining of Massive Datasets." Cambridge University Press, 2011.
[24] François Rousseau, Emmanouil Kiagias, and Michalis Vazirgiannis. "Text categorization as a graph classification problem." In *ACL*, vol. 15, pp. 1702-1712, 2015.
[25] A. Reyes, P. Rosso. "Making objective decisions from subjective data: Detecting irony in customer reviews." *Decision Support Systems*, vol. 53, no. 4, pp. 754-760, 2012.

[26] Y. Kim. "Convolutional neural networks for sentence classification." arXiv preprint arXiv: 1408.5882, 2014.
[27] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. "Natural Language Processing (Almost) from Scratch." *J. Mach. Learn. Res.* vol. 12, pp. 2493–2537, 2011.
[28] Yarin Gal and Zoubin Ghahramani. "A Theoretically Grounded Application of Dropout in Recurrent Neural Networks." In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems*, Barcelona, Spain, pp. 1019–1027, 2016.
[29] Mikolov, Tomas & Sutskever, Ilya & Chen, Kai & Corrado, G.s & Dean, Jeffrey. "Distributed Representations of Words and Phrases and their Compositionality." *Advances in Neural Information Processing Systems*. vol. 26, pp. 3111–3119, 2013.
[30] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. "Glove: Global vectors for word representation." In *EMNLP*, 2014.
[31] K Cho, B van Merrienboer, C Gulcehre, F Bougares, H Schwenk, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation", *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Qatar, 2014.
[32] R. Socher, B. Huval, C. D. Manning, and A. Y.Ng, "Semantic compositionality through recursive matrix-vector spaces", Proceedings of the *2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, pp. 12011211, 2012.
[33] Tai, Kai Sheng, Richard Socher and Christopher D. Manning. "Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks." ArXiv abs/1503.00075, 2015.
[34] Mikael Henaff, Joan Bruna, and Yann LeCun. "Deep Convolutional Networks on Graph-Structured Data", CoRR, 2015.
[35] T. Y. Liu, Y. Yang, H. Wan, H. Zeng, Z. Chen, and W. Y. Ma. "Support vector machines classification with a very large-scale taxonomy." *ACM SIGKDD Explorations Newsletter*, vol. 7, no. 1,pp. 36–43, 2005.
[36] Gui-Rong Xue, Dikan Xing, Qiang Yang, and Yong Yu. "Deep classification in large-scale text hierarchies." In *SIGIR*. pp. 619–626, 2008.
[37] Johnson R, Zhang T. "Effective use of word order for text categorization with convolutional neural networks." In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2015)*, 2015.
[38] Tang D, Qin B, Liu T. "Document modelling with gated recurrent neural network for sentiment classification." In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, 2015.
[39] Dou ZY. "Capturing user and product Information for document level sentiment analysis with deep memory network." In *Proceedings of the Conference on Empirical Methods on Natural Language Processing (EMNLP 2017)*, 2017.
[40] Zhou X, Wan X, Xiao J. "Attention-based LSTM network for cross-lingual sentiment classification." In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, 2016.
[41] Kim Y. "Convolutional neural networks for sentence classification." In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, 2014.
[42] Wang J, Yu L-C, Lai R.K., and Zhang X. "Dimensional sentiment analysis using a regional CNN-LSTM model." In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, 2016.
[43] Guggilla C, Miller T, Gurevych I. "CNN-and LSTM-based claim classification in online user comments." In *Proceedings of the International Conference on Computational Linguistics (COLING 2016)*, 2016.
[44] Yu J, Jiang J. "Learning sentence embeddings with auxiliary tasks for cross-domain sentiment classification." In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, 2016.
[45] Ben Ismail, Mohamed Maher & Bchir, Ouiem. "Insult Detection in Social Network Comments Using Possibilistic Based Fusion Approach." *Springer International Publishing*, pp.15-25, 2014.
[46] Yann LeCun, Leon Bottou, Genevieve B. Orr and Klaus-Robert Muller. "Efficient backprop. Neural networks: Tricks of the trade." *Springer Berlin Heidelberg*, pp. 9-48, 2012.
[47] John Duchi, Elad Hazan, and Yoram Singer. "Adaptive subgradient methods for online learning and stochastic optimization." *The Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.
[48] Joulin, Armand, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hervé Jégou and Tomas Mikolov. "FastText.zip: Compressing text classification models." ArXiv abs/1612.03651, 2016.
[49] Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, RonWeiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. "Scikit-learn: Machine learning in python." *Journal of Machine Learning Research*, vol. 12, no. 10, pp. 2825–2830, 2012.