Spam Filtering in SMS using Recurrent Neural Networks

Rahim Taheri
PhD Student
Department of Computer
Engineering and IT
Shiraz University of Technology
Shiraz, Iran
Email: Tahery.Rahim@gmail.com

Reza Javidan
Associate Professor
Department of Computer
Engineering and IT
Shiraz University of Technology
Shiraz, Iran
Email: Reza.javidan@gmail.com

Abstract— **Short Message Service (SMS) is one of the mobile communication services that allows easy and inexpensive communication. Producing unwanted messages with the aim of advertising or harassment and sending these messages on SMS have become the biggest challenge in this service. Various methods have been presented to detect unsolicited short messages; many of which are based on machine learning. Neural Networks have been applied to separate the unwanted text messages (known as spam) from normal short messages (known as ham) in SMS. To the best of our knowledge, Recurrent Neural Network (RNN) has not been used in this issue yet. In this paper, we proposed a new method which utilizes RNN to separate the ham and spam with variable length sequences; even though we used a fixed sequence length. The proposed method achieved an accuracy of 98.11, indicates a considerable improvement compared to Support Vector Machine (SVM), token-based SVM and Bayesian algorithms with accuracies of 97.81, 97.64, and 80.54, respectively.**

Keywords- Prediction; RNNs; SMS; Spam; Ham

I. INTRODUCTION

In recent years, the use of mobile phones and smart systems has increasingly developed, and Short Message Service (SMS) has become one of the most important means of communication; so that 97% of cell phone users use this service [1]. Unwanted short messages (known as spams) are transferred on the communication channel such as SMS, perform great advertising, but as a disturbing factor for users. In 2012, more than 6 billion messages were transferred on mobile phones in USA [2]. According to a study in 2011, 3.5 billion people or 80% of active users in the world use SMS of mobile phone as a means of communication. Of this large number of short messages, many are unwanted SMS messages produced for the following reasons.

- Sending SMS is low cost and many mobile operators offer SMS packages with very low price.
- Since users interact more with mobile phones compared to computers, they have more confidence in SMS, and it is very convenient to send confidential information [3].

According to some studies conducted in 2008, mobile phones receive more spam than normal SMS. Similarly, some statistics in America showed that mobile phone users receive 1.1 billion spams annually, of which include about 44% of mobile devices [4]. Likewise, Chinese users receive 8.29 spams per week [5]. Moreover, according to research in 2011, the issue of spam is a core issue in the Middle-East and South-East Asia that includes 20 to 30% of total traffic.

There are two main types of methods to detect unwanted SMS: techniques along with user participation and content-based methods. Methods based on user participation are based on feedback from users and sharing their experiences. Because of the problems associated with data access and user experience, these methods are rarely used, but content-based methods act based on content analysis of text messages and are more common.

High volume of SMS traffic has provided an opportunity for creators of spam, so that instead of creating spam in email, create it in SMS [2]. Previous research shows that a highly effective method used in filtering unwanted SMS is risk analysis in message content [6]. The problem of unwanted short message in SMS is very similar to filtering these messages in an email. However, there are cases related to the nature of SMS:

- Short messages contain up to 160 characters.
- Users use a subset of the words, including slang, Internet abbreviations, and short text [2,7]. Therefore, using methods that consider these differences seems essential.

This paper has predicted unwanted SMS messages accurately using Recurrent Neural Network (RNN). Innovations in this paper are the:

- Use of methods that apply preprocessing actions as a part of algorithm.
- Show higher accuracy with the increase in the number of layers of neural networks used.
- Moreover, the recursive nature of this algorithm has made it suitable for sequential data and effective in achieving high accuracy of the algorithm.

978-1-5386-2585-9/17/$31.00 ©2017 IEEE

This paper has uses RNNs to predict whether short messages are unwanted or normal. The architecture used in this paper is shown in Figure 1. It consists of a sequence of inputs from a marked text to RNN, and classifies the last output of RNN according to its zero or one value as an unwanted or normal text message. After entering the tokens to the initial state, they are applied to central functions of hidden layer, and determined in the last layer of the intended output, which is in fact the class related to entering data.

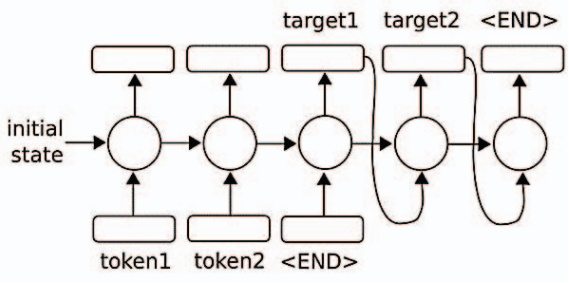

Figure 1: To predict a single number or a group, we consider a sequence of inputs and obtain the final output as predicted output.

Based on RNN, the proposed method in this paper achieved an accuracy of 98.11, which suggests a considerable improvement compared to the SVM, token-based SVM, and Bayesian algorithms, which were used in the most recent studies, with accuracies of 97.81, 97.64, and 80.54, respectively.

The rest of this paper is organized as follows. In Section 2, we examine the prior research in predicting the unwanted SMS messages. Section 3 describes the proposed method. Section 4 and Section 5, respectively, include results, and discussion and conclusions.

II. Related work

Techniques based on machine learning require a sufficient amount of training data used for categorizing labeled models. Since designing a suitable text-classification technique has many problems, some monitored methods, such as Bayes classifier, nearest neighbor, SVM, C4.5 decision trees, and neural networks are used for this problem. Ahmed et al. have offered a monitoring strategy that has used Bayes algorithms with Appriori for classification of short messages [8]. Nejadet et al. have offered a new classification algorithm that uses the combination of classification algorithms without change in the original algorithm to obtain better performance [9].

A variety of clustering techniques are used for the detection of unwanted short message in SMS. One of the first studies in this regard concerns using SVM. Gomez et al. used a similar classification method that uses Information Gain (IG) to choose the marks [10]. Logzhen et al. used a k-nearest neighbor algorithm along with other methods of detection of unwanted short message on a data set containing 750 cases of unwanted and normal short messages [11]. Liu and Wang used the recurrence of some text units as input parameters to Bayesian classification algorithms, k-nearest neighbors, SVM and so on [12]. Almeida et al. tested 13 classification algorithms on a data set of more than 5500 SMS (747 unwanted short message and 4827 normal SMS). Their results show that SVM with alphabetic marking has the best performance [13].

Alphabetic marking includes separating alphabetic and non-alphabetic characters. Finally, they extracted 8100 marks from SMSs. Delaney et al. [14] changed the methods and data sets used in the study by Almeida et al. This study focused on factors of assessment of categories of unwanted SMS messages. The accuracy of the proposed method shows that their method does not work well for ordinary short messages.

In summary, reviewing the studies conducted shows that methods based on SVM and Bayesian are more popular methods for classifying unwanted SMS messages. The best performance reported in papers regarding detecting unwanted short messages is an accuracy of 97.59% achieved using 81000 features. In this paper, by using RNNs to identify unwanted SMS messages, we have obtained higher accuracy of 98.11%. The main reason is that, in practice, the RNN actually runs computationally faster and that, if we choose, we can run sequences of different lengths through the RNN.

III. The Proposed Method

The ability of the brain to process huge volumes of information in a short time, the use of parallel structure in data analysis, and the remarkable ability of the human brain in learning various issues are special features. Therefore, simulation is always been tempting, and deep neural networks have been created for this purpose. Among all machine learning algorithms, Deep Recurrent Neural Networks (DRNNs) work on data sequence [15]. Sequential data are data whose current values depend on previous values [16]. Among such data, the following can be noted.

- Frames (samples) of speech signal
- Continuous Frames (images) of Video
- Climatic condition
- The stock price of a company / industry
- The sequences generated by the grammar
- Words within a text [15]

RNNs are very powerful because of combining the following features:

- Distributed hidden layers, which allow them to save a lot of information about previous layers.
- Non-linear dynamics, which allows such networks to update hidden layers in a complex way.
- RNNs have the potential to provide implementation and enforcement for small and parallel programs, and thus have a great interaction for producing more complicated results.
- With the number of neurons at hand and enough time, RNNs have the ability to do any calculations performed by a computer.

Nevertheless, high computational power of RNNs has made training them very difficult, and deepening of this type of network and the ability for parallel processing have solved this problem to some extent [17,18].

A. Recurrent Neural Network (RNN)

A RNN contains an input layer, hidden layer, and output layer, as well as feedback connection weights, activation functions, and interconnection weights. In this study, the proposed RNN is designed by the combination of the locally recurrent and globally feed forward structure [18]. The dynamic properties are achieved by utilizing the internal feedbacks as shown in Figure. 2.

For a clear understanding of the computational model for the proposed RNN through the proposed structure, the mathematical function of each node is described as follows:

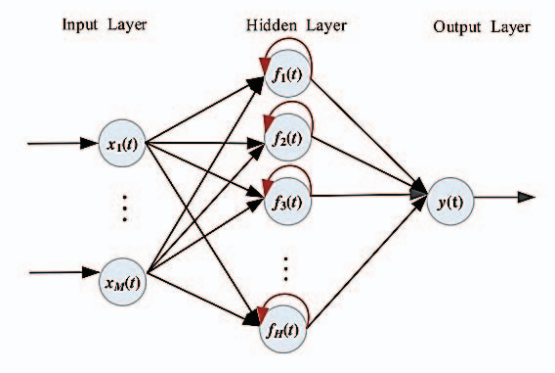

Figure 2: The structure of recurrent neural network

Input layer: There are M nodes, which represent the input variables in this layer. The output values of each node can be described as Equation (1):

$$u_i(t) = x_i(t), \quad (1)$$

where $u_i(t)$ is the ith output value at time t, i = 1, 2, . . ., M, and the input vector is $x(t) = [x_1(t), x_2(t), \ldots, x_M(t)]$.

Hidden layer: each node in hidden layer connects with the input nodes and output node of RNN. The output of each hidden node is shows in Equation (2):

$$v_j(t) = f\left(\sum_{i=1}^{M} w_{ij}^1(t)u_i(t) + v_j^1(t)\right), (i = 1, 2, \ldots, M; j = 1, 2, \ldots, H), \quad (2)$$

where $f(x) = (1 + e^{-x})^{-1}$ is the activation function, w1 ij(t) is the input weight connecting the ith node in the input layer with the j^{th} hidden node, H is the number of hidden nodes, and v1 j(t) is the feedback value, which is given by Equation (3):

$$v_j^1(t) = w_j^2(t)v_j(t-1), \quad (3)$$

where $v_j(t - 1)$ is the output value of the hidden layer at time t - 1, $w^2_j(t)$ is the self-feedback weight of the jth hidden node.

Output layer: There is only one node in this layer, the output is computed by Equation (4):

$$y(t) = \sum_{j=1}^{H} w_j^3(t)v_j(t), \quad (4)$$

where y(t) is the output value of output layer at time t, $w^3_j(t)$ is the output weight of the j^{th} hidden node.

Moreover, for estimating the performance, the root mean squared error (RMSE) is defined by Equation (5) [19]:

$$E(t) = \sqrt{\frac{1}{2t}\sum_{p=1}^{t}(y_d(p) - y(p))^2}, \quad (5)$$

where y(p) and $y_d(p)$ are the network output and the desired output at time p (p = 1, 2, . . ., t) [20].

B. Proposed Structure

An RNN is the generalization of feed-forward neural networks used for data with sequence. In this paper, a modified version of the standard RNN, which was described in the earlier section, was used with the following assumptions and alterations:

According to Equation (6), RNNs convert a sequence of input data (x_1.....x_t) to a sequence of data with hidden items (h_1....h_t) and a sequence of data with hidden status sequence to output data (y_1.....y_t) makes the following.

$$h(t) = f_H(W_{IH}x(t) + W_{HH}h(t-1)) \quad (6)$$
$$y(t) = f_o(W_{HO}h(t))$$

In Equation (6), f_H and f_o are non-linear functions such as sigmoid or tanh, and x(t) is a vector with fixed length as a demonstration of entrance. h(t) represents the hidden mode, W_{ij} represents the weights that connect layers of neurons to each other, and y(t) is the output vector [18]. Neural networks that have recurring edges in their structure, unlike feed-forward neural networks, the edges can form a cycle. Because of having recursive edge, Figure 3 shows the quality of using RNNs in this paper. This type has memory in their structure and is suitable for sequential data processing [17].

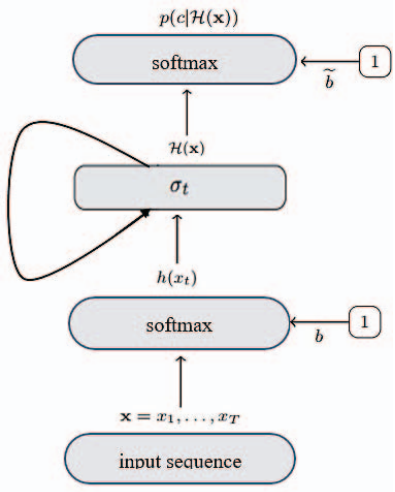

Figure 3: RNN structure with the return of the hidden neurons

Suppose that according to Equation (7), we assume a fully connected neural network.

$$y = \sigma(Ax) \quad (7)$$

Here, weights give the value of y in the output layer by multiplying A in the input layer x and then running it by activating σ. If we have a sequence of data x_1, x_2, x_3, x_4, and so on, we can build fully connected layer with the help of Equation (8) based on previous and entrances layers.

$$y_t = \sigma(By_{t-1} + Ax_t) \quad (8)$$

On top of this recursive iteration, we want to consider the outputs of probability distribution function to the next stage entrance, so we will have, Equation (9).

$$s_t = \text{softmax}(Cy_t) \quad (9)$$

The regression function, we have used softmax in the study that uses the Equation (10) [18].

$$\sigma(z)_j = \frac{e^{z_j}}{\Sigma_{k=1}^{K} e^{z_k}} \quad (10)$$

Equation (10) is used both about the entry layer into hidden layers, within hidden layers, and between hidden layer and about output layer. When we have an exit with full sequence $\{s_1, s_2, s_3, ...\}$, we can determine the target by looking at a number or a category.

IV. Results and Discussion

There are few data sets to predict spam in short message service systems, among which UCI datasets are used in most studies. In this study, we have used RNNs as an appropriate classifier algorithm on this dataset. Thus, in this section of the paper, we first introduce this data set briefly. Then we explain the quality of performing the experiments, where the results on this data set in classification show satisfactory improvement.

A. Data set used

The dataset used in this study are of dataset prepared in UCI known as UCI SMS Spam [21], and the dataset includes 5574 SMS labeled classified into two groups: 747 SMS messages are unwanted and 4827 SMS messages are normal SMS messages or an SMS message with the user's consent. As it can be seen in Table 1, and Table 2, some of the statistical characteristics of the messages in the dataset employed in this paper are shown.

Table 1: Profile of UCI SMS Spam dataset

	Frequency	Percent
ham	4827	86.6%
spam	747	13.4%
Total	5574	100%

Table 2. Statistical characteristics of the messages in the UCI dataset.

count	5574
mean	80.48
Standard Deviation	59.94
Min length	2
Max length	910

The number of characters and frequency of messages in both ham and spam message types are shown in figure 4 and Figure 5, respectively.

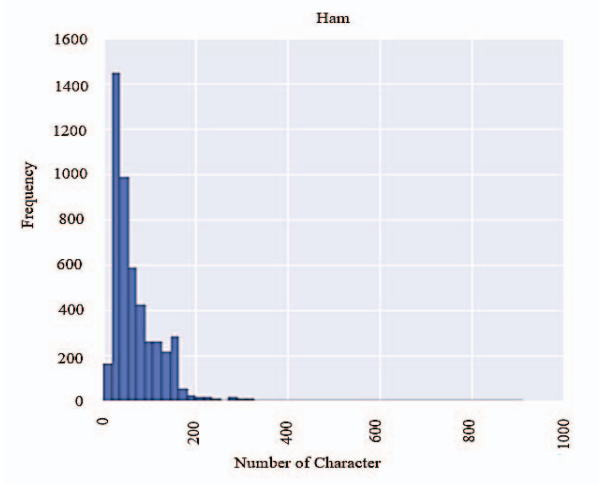

Figure 4. The number of characters of messages with respect to the message type in the UCI dataset.

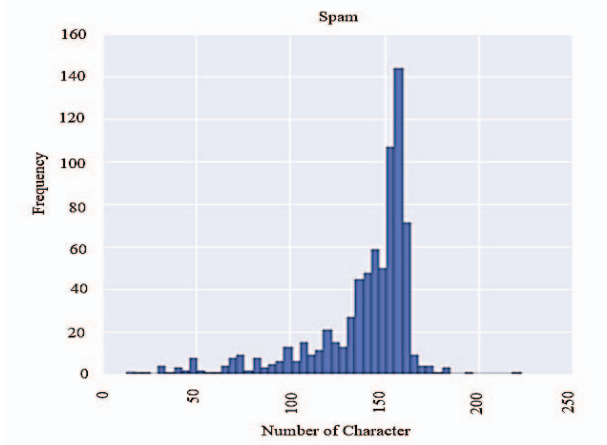

Figure 5. The number of frequency of messages with respect to the message type in the UCI dataset.

B. Implementation details

For the implementation of the proposed method, similar to many machine learning algorithms, we have considered a training algorithm, a cycle or a period of applying algorithm on all training vectors which called epoch. The proposed method has been implemented for a number of different epochs listed in the results. The parameters considered in this study are shown in Table 3. One method which is used to increase the accuracy is batching data. In this paper, it was chosen as 25. This means that in each entry in a batch with 25 sequences of words, and the maximum size of each word sequences, we have adopted to be 25. In this paper, we have considered 100 for rnn_size, and each word in a training vector got a size of 50. Learning rate considered in this article is 0.0005. In the simulations carried out in this paper, 70% of the data, i.e. 3901 records as training and 30% of data, equivalent to 1673, as the test are used.

Table 3: Parameters and values considered in simulation

The intended parameters	Values considered
batch_size	25
max_sequence_length	25
rnn_size	100
embedding_size	50
min_word_frequency	10
learning_rate	0.0005
Training Data	70%
Test Data	30%

C. *Analysis of the results*

The first step, we ran simulations for 200 epochs. As it can be seen in Figure 6, the prediction accuracy in the numbers 103 and 116 has reached 98.11%, so naturally in these two implementations, softmax function has its minimum value i.e. 0.34.

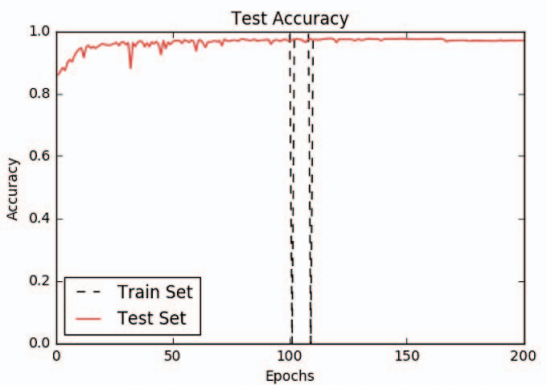

Figure 6: Accuracy of implementation of prediction algorithm for 200 epochs

In the next step, we ran simulations for 1000 epochs. As can be seen in Figure 7, the prediction accuracy in different implementations has reached 98.11%. We observe after reaching this steady state in the next epochs, high 98% accuracy rate has remained fairly stable and the results have nothing to do with the number of additional epochs.

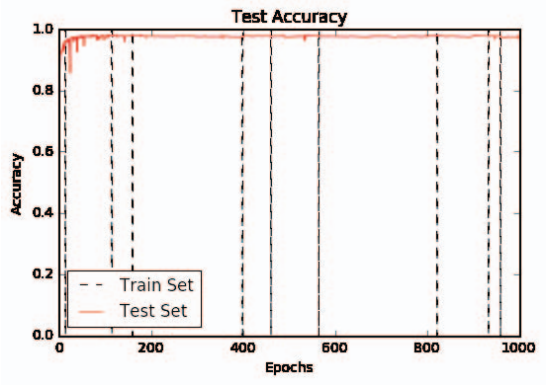

Figure 7: Accuracy of implementation of the prediction algorithm for 1000 epochs

The result of implementation of the proposed algorithm for 2000 epochs can be seen in Figure 8. With this number of epochs the results confirmed that after reaching this steady state in the next epochs, high 98% accuracy rate has remained fairly stable and the results have nothing to do with the number of additional epochs.

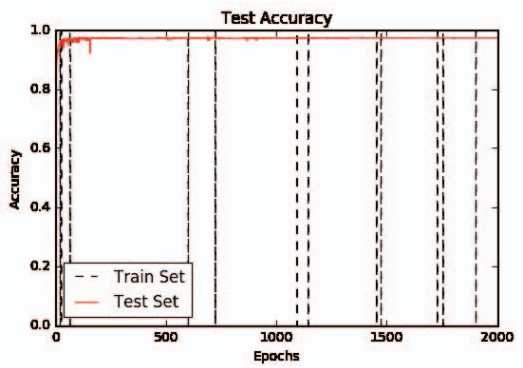

Figure 8: Accuracy of the prediction algorithm for 2000 epochs

The simulation results for the proposed method for different number of epochs was demonstrated in Figures 6 to 8. As shown, after reaching the steady state, the accuracy of the algorithm is not significantly changed, confirming that no overfitting occurs.

In comparing the proposed method with other methods offered in previous researches, it can be seen that the proposed method is more accurate. The accuracy of the proposed method is compared to three of the most recent studies in Table 4. This accuracy was obtained for a 200-epoch run of the algorithm.

Table 4: Comparing the accuracy of spam prediction algorithms

Method	Accuracy (%)
Proposed Approach	98.11
SVM-based spam filter [22]	97.81
NB-based spam filter [22]	80.54
SVM + tok1[23]	97.64

Table 5: Comparing the execution time of spam prediction algorithms

Method	Execution time(second)
Proposed Approach (200 epoch)	304
SVM-based spam filter [22]	263
NB-based spam filter [22]	235
SVM + tok1[23]	256

As shown in Figure 9, the accuracy of the proposed method is improved compared to the other two algorithms. The runtimes of the algorithms are compared in Table 5, according to which, the runtime of the proposed algorithm is negligibly longer than the other three. The runtime of the proposed method in Table 5 was calculated for 200 epochs, ultimately achieving an accuracy of 98.11.

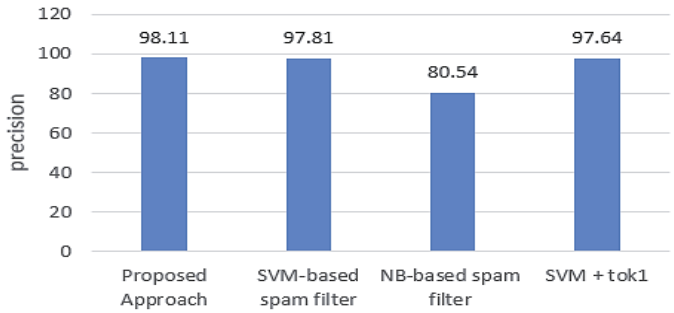

Figure 9: Comparing the accuracy of spam prediction algorithms

V. CONCLUSION

In this paper, a classification method is provided for detecting unwanted and normal messages. So far, in the previous studies, SVM or Bayesian methods is used. The proposed method RNNs is used. Test results on standard datasets UCI SMS spam showed the efficiency of the proposed method. In the proposed method, after 100 initial epochs, when a steady state is observed, a high accuracy of 98% is obtained has not achieved so far in any other researches, and increase the number of epochs of the algorithm implementation will not impact of the result.

The proposed method in this paper considers the pre-processing stage part of the classification algorithm and offers a higher accuracy compared to the most recent studies. Moreover, the proposed method can be used an appropriate alternative to the previous methods considering its acceptable runtime.

REFERENCES

[1] Al Moubayed N., Breckon T., Matthews P., McGough A.S. (2016) SMS Spam Filtering Using Probabilistic Topic Modelling and Stacked Denoising Autoencoder. In: Villa A., Masulli P., Pons Rivero A. (eds) Artificial Neural Networks and Machine Learning ICANN 2016. Lecture Notes in Computer Science, vol 9887. Springer, Cham

[2] Karami, Amir and Zhou, Lina. (2014) Improving static SMS spam detection by using new content-based features, Twentieth Americas Conference on Information Systems, Savannah

[3] Ishtiaq Ahmed and Rahman Ali and Donghai Guan and Young-Koo Lee and Sungyoung Lee and TaeChoong Chung. (2015) Semi-supervised learning using frequent itemset and ensemble learning for *f*SMS*g* classification, Expert Systems with Applications, Volume 42, Pages 1065 – 1073

[4] P. He and Y. Sun and W. Zheng and X. Wen. (2008) Filtering Short Message Spam of Group Sending Using CAPTCHA, First International Workshop on Knowledge Discovery and Data Mining (WKDD 2008), Pages 558-561

[5] Q. Wang and X. Han and X. Wang. (2009) Studying of Classifying Junk Messages Based on The Data Mining, International Conference on Management and Service Science, IEEE, Pages 1-4

[6] Sebastian, Libina Rose and Jacob, Smitha and Thomas, Teena. (2015) Handling Different Types of Messages in On-Line Social Network using Naïve Bayes, International Journal for Innovative Research in Science & Technology, Volume 5, Pages 204–207

[7] Thiago S. Guzella and Walmir M. Caminhas. (2009) A review of machine learning approaches to Spam filtering, Expert Systems with Applications, Volume 36, Pages 10206 – 10222

[8] Ahmed, Ishtiaq; Guan, Donghai; Chung, Tae Choong. (2014) Sms classification based on naive bayes classifier and apriori algorithm frequent itemset, International Journal of Machine Learning and Computing; Singapore4.2: Pages 183-187.

[9] Najadat, Hassan and Abdulla, Nawaf and Abooraig, Raddad and Nawasrah, Shehabeddin. (2014) Mobile sms spam filtering based on mixing classifiers, International Journal of Advanced Computing Research, Volume 1, Pages 1 – 7

[10] Gomez Hidalgo, Jos ´ e Mar ´ ´ıa and Bringas, Guillermo Cajigas and Sanz, Enrique Puertas and Garc ´ ´ıa, Francisco Carrero. (2006) Content based SMS spam filtering, Proceedings of the 2006 ACM symposium on Document engineering, ACM, Pages 107 – 114

[11] Duan, Longzhen and Li, Nan and Huang, Longjun. (2009) A new spam short message classification, First International Workshop on Education Technology and Computer Science, 2009. IEEE., Volume 2, Pages 168 – 171

[12] Liu, Wuying and Wang, Ting. (2010) Index-based online text classification for sms spam filtering, Journal of Computers, Academy Publisher, PO Box 40 Oulu 90571 Finland, Volume 5, Pages 844 – 851 Predicting SPAMs in Short Message Service using Recurrent Neural Networks 9

[13] Almeida, Tiago A and Hidalgo, Jose Mar ´ ´ıa G and Yamakami, Akebo. (2011) Contributions to the study of SMS spam filtering: new collection and results, Proceedings of the 11th ACM symposium on Document engineering, ACM, Pages 259 – 262

[14] Delany, Sarah Jane, Mark Buckley, and Derek Greene. (2012) "SMS spam filtering: Methods and data." Expert Systems with Applications 39. Volume 10: Pages 9899-9908.

[15] Su, Bolan and Lu, Shijian. (2017) Accurate recognition of words in scenes without character segmentation using recurrent neural network, Pattern Recognition, Elsevier, Volume 63, Pages 397- 405

[16] Venugopalan, Subhashini and Xu, Huijuan and Donahue, Jeff and Rohrbach, Marcus and Mooney, Raymond and Saenko, Kate. (2014) Translating videos to natural language using deep recurrent neural networks, arXiv preprint arXiv:1412.4729

[17] Guo, Liang and Li, Naipeng and Jia, Feng and Lei, Yaguo and Lin, Jing. (2017) A recurrent neural network based health indicator for remaining useful life prediction of bearings, Neurocomputing, Elsevier, Volume 240, Pages 98- 109

[18] Chen, Yu and Yang, Jian and Qian, Jianjun. (2017) Recurrent neural network for facial landmark detection, Neurocomputing, Elsevier, Volume 219, Pages 26- 38

[19] Pajouh, Hamed Haddad, et al. "A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IoT backbone networks." IEEE Transactions on Emerging Topics in Computing (2016).

[20] Han, Hong-Gui, et al. (2016) "A soft computing method to predict sludge volume index based on a recurrent self-organizing neural network." Applied Soft Computing, Elsevier, Volume 38: Pages 477-486.

[21] http://archive.ics.uci.edu/ml/datasets

[22] Aragao, Marcelo VC and Frigieri, Edielson Prevato and Ynoguti, Carlos A and Paiva, Anderson P. (2016) ˜ Factorial design analysis applied to the performance of SMS anti-spam filtering systems, Expert Systems with Applications, Elsevier, Volume 64, Pages 589-604

[23] El-Sayed M. El-Alfy, Ali A. AlHasan. (2016) Spam filtering framework for multimodal mobile communication based on dendritic cell algorithm, Future Generation Computer Systems, Volume 64, Pages 98-107

[24] Jain, G., and B. Manisha. "Spam Detection on Social Media Text." (2017): 63-70