The Impact of Deep Learning Techniques on SMS Spam Filtering

Wael Hassan Gomaa
Faculty of Computers and Artificial Intelligence
Beni-Suef University, Beni-Suef, Egypt, Shaqra University
Kingdom of Saudi Arabia

Abstract—**Over the past decade, phone calls and bulk SMS have been fashionable. Although many advertisers assume that SMS has died, it is still alive. It is one of the simplest and most cost-effective marketing tools for companies to communicate on a personal level to their customers. The spread of SMS has led to the risk of spam. Most of the previous studies that attempted to detect spam were based on manually extracted features using classical machine learning classifiers. This paper explores the impact of applying various deep learning techniques on SMS spam filtering; by comparing the results of seven different deep neural network architectures and six classifiers for classical machine learning. Proposed methodologies are based on the automatic extraction of the required features. On a benchmark data set consisting of 5574 records, a fabulous accuracy of 99.26% has been resulted using Random Multimodel Deep Learning (RMDL) architecture.**

Keywords— SMS Spam Filtering; Deep Learning; RNN; GRU; LSTM; CNN; RCNN; RMDL

I. Introduction

Telephone calls and mass text messages have been common over the past decade. For businesses to connect with their consumers on a personal level, it is one of the best and most cost-effective marketing tools. Internet and social media also reward the media, but SMS text messages are instant means that need few barriers to reach your audience with your information. Given that by 2020, seven out of every 10 people will own a smartphone, it will need to be easy to use and non-intimidating for any technology-based solution to achieve critical mass via digital. When companies want to keep up with this growing community, they need to continue looking for ways to make their company more mobile-friendly. In three seconds, more than 90% of people read an SMS, the chances of readability are extremely high. SMS may help increase customer engagement, promote products and services, or provide viewers with urgent updates. SMS messaging is also designed to enable companies to send messages to any contact number worldwide from any website or service using an API. Thanks to its omnipresent nature, SMS spamming has become a major nuisance for mobile subscribers. It includes substantial costs in terms of lost productivity, use of network bandwidth, administration, and personal privacy attack. Previous methods that detect SMS spam are hampered by the limitations of manual extraction, which is not efficient enough .Previous experiences and domain expertise are needed to identify prospective features for appropriate classification. Even then, it is important to reassess the selection of features on the basis of certain parameters, such as information acquirer. Deep learning is a category of artificial intelligence machine learning mechanisms. Deep neural network is a very effective way to avoid the wasteful phase of feature selection and extraction. For automatic pattern recognition and unsupervised feature learning, multiple layers of data are used. In order to automatically extract essential features and eliminate classification errors, the deep neural network components act with each other to train themselves sequentially. The effect of applying different deep learning techniques is discussed in this paper by comparing the accuracy of seven different deep neural network architectures and six classifiers for classical machine learning. The experiments are performed using Keras API and RapidMiner platform on a popular UCI benchmark dataset. The rest of the paper is structured as follows, Section II presents the related work, Section III describes the suggested methodologies, Section IV presents the experiments and findings and finally Section V outlines conclusion and future work stages.

II. Related Work

Several articles reviewed the previous and current approaches for SMS spam according to various metrics [1-7]. The most detailed and valuable review was presented in [1], it covers most of the SMS applications, approaches and methods. The authors in [1] searched in more than ten scientific databases such as Springer, IEEE, ACM and Google Scholar. They applied seven search strategies: approach-based, architecture-based, source-based, method-based, corpus-based, status-based and application-based. They found more than one thousand related articles; to exclude the obsolete and redundant papers, several sifting criteria were performed. For further research, eighty-three related articles were finally chosen. SMS spam detection methods are divided into three categories: machine learning, statistical analysis, and evolutionary algorithms. Naïve Bayes and decision tree are samples of machine learning algorithms. Statistical analysis techniques include models focused on mathematics, like factor analysis. Evolutionary algorithms are based on textual contents and biological techniques like genetic algorithms. The presented papers were categorized into three categories, content-oriented, non-content-oriented, and hybrid, according to approaches. The selected articles are classified into three categories according to architectures: client, server and collaborative. This systematic analysis presented many useful and interesting results for SMS spam filtering and concluded that much research remains to be done to improve existing approaches and methods [1].

There was another important review in [2]. The researchers in [2] presented a description of the problems, strategies and opportunities currently available on the role of SMS spam filtering. They introduced taxonomy of two main classes of methods and techniques: the access layer (AL) and the service provider layer. There are six sections in each class: Writing Style, Bayesian Network, SVM-based, Machine Learning, Evolutionary Algorithms, and Other Techniques. They compared 51 references in terms of datasets descriptions, proposed techniques, comparison techniques and major findings. They also concentrated on the weaknesses of existing studies and pointed out the paths for future research. Authors in [3] considered 17 papers and reviewed their algorithms, approaches, used databases, benefits, drawbacks, and methods of evaluation. In addition, they explained the classical machine learning classification problems. They concluded that the problems of shortcut terms and regional content were not addressed by any study. Authors in [4] concentrated on the various data mining strategies for the detection of SMS spam. Authors in [5] summed up recent efforts to reduce spam on SMS. In order to achieve better results, they recommended using the Support Vector Machine (SVM) classifier. Authors in [6] contrasted the algorithms, strengths, weakness, number of features, reliability and the data set used between 16 papers. Eventually, the first systematic review describing and comparing the major SMS spam identification processes, architectures and approaches was published in [7].

In addition, some research has been presented in [8-11] on deep learning approaches for the detection of SMS spam. Using text information only, CNN and LSTM were tested in [8]. On both balanced and imbalanced datasets, the experiments were performed. The results obtained indicated that the architecture of the CNN was superior to the architecture of the LSTM. Three models are presented in [9]: RNN, LSTM, and Semantic LSTM (SLSTM). SLTSM uses the semantic layer on top of the LSTM. The semantic layer is built using ConceptNet, WordNet, and Word2Vec.In [10] preprocessing of dataset was applied through stemming, sentiment analysis, stop word removing and tokenization. To be an input to CNN, a matrix of TF-IDF features was created. Authors in [11] tested CNN on two datasets containing respectively 5574 and 2000 SMS. The results of our suggested methodologies will be compared to the four researches mentioned above.

III. Proposed Methods

A. *Classical Machine Learning Classifiers*

In this section we discuss briefly the six used machine learning classifiers: Naïve Bayes, Generalized Linear Model (GLM), Fast Large Margin, Decision Tree, Random Forest, Gradient Boosted Trees and Support Vector Machine. A Naive Bayes classifier is a framework used for probabilistic machine learning classification tasks. It's easy to use but computationally cost effective. Naive Bayes' fundamental assumption is that, given the tag (class) value, the value of any attribute is independent of the value of any other attribute. GLM estimates models of regression for results after exponential distributions. These include the Poisson, binomial, and gamma distributions in addition to the Gaussian distribution. Each serves a different purpose and can be used either for prediction or classification depending on the choice of distribution and connection function. GLM is considered a dynamic version of linear regression models. Fast Large Margin is an SVM-Like algorithm which runs in O(N). Because of its complexity, Fast Large Margin is perfect for classifying big data. A decision tree is a flowchart-like architecture; it can be used to represent decision and decision-making visually and clearly. Each part of decision tree has a role in the classification process; the inner nodes represent checking of attributes, edges represent result of checking and the terminal nodes represent class labels. Random Forest model is developed from decision trees. The primary idea of a random forest is merging a number of decision-making trees into one model. Separately, the findings of decision trees may led to non-perfect results, but in combination, the findings are enhanced. Gradient Boosted Trees have the same idea of Random Forest models; but the difference is that in Gradient Boosted models the combination task starts at the beginning. If the parameters are carefully adjusted, gradient models may produce better results than random forests. The disadvantage of gradient models is that they suffer from noisy data. SVM is a popular and simple supervised classifier that depends on finding the hyper-plane which makes two given categories somewhat different. SVM is efficient in situations where the number of dimensions exceeds the number of instances.

B. *Deep Learning Techniques*

- Deep Neural Network (DNN): Older neural network models such as the first perceptrons were small, consisting of single input and single output layer, and at most single hidden layer between them. More than three layers count as "deep" learning (including input and output). It is a concept that is narrowly defined, meaning more than one hidden surface as shown in Fig. 1. Every layer of nodes trains in deep-learning networks on a different group of features according to the execution of preceding layers. The further you go into the neural net, the more complex the characteristics the nodes will identify as they integrate and recombine characteristics from previous layer.

- Recurrent Neural Network (RNN): RNN is a kind of artificial neural network with a "memory" that saves the previous information that is needed. The key aspect of RNN is Hidden State, which recalls certain knowledge about any given sequence such as word set in a sentence. The idea of RNN is to create many copies of the same architecture, each transmitting data to the next network, as shown in Fig. 2. Unlike other neural networks, it reduces the complexity of the parameters. RNN has many benefits, but suffers from the issue of vanishing gradient.

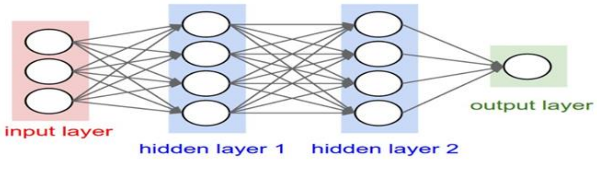

Fig. 1. Deep Neural Network.

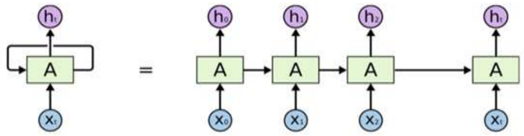

Fig. 2. Architecture of RNN.

- Long Short-Term Memory (LSTM): Several variants have been developed to resolve the problem of Gradients vanishing in RNN.LSTM is considered the best of them. In theory, a repeating LSTM system tries to "remember" all past information that the network has so far been seen and to "forget" irrelevant data. This is accomplished by adding different layers of activation functions called "gates" for different purposes. Each recurrent LSTM unit also preserves a vector called the Internal Cell State that determines the information that the previous recurrent LSTM unit has chosen to maintain conceptually. As shown in Fig. 3, LSTM contains four different gates: Input, Output, Input Modulation and Forget Gate.

- Gated Recurrent Unit (GRU): GRU is intended for solving the problem of the vanishing gradient that comes with a regular recurrent neural network. GRU is viewed as a variant of the LSTM. They have similar structures and deliver equally excellent results in some cases. This consists of only three gates, unlike LSTM, and does not retain an Internal Cell State. The data that is contained in an LSTM recurrent unit in the Inner Cell State is inserted into the Gated Recurrent Unit's secret state. The shared data will be passed to the next Gated Recurrent System. As shown in Fig. 4, Update Gate, Reset Gate, and Current Memory Gate are the different gates of a GRU.

- Convolutional Neural Network (CNN): Originally designed to conduct deep learning for computer vision tasks, the Convolutional Neural Network (CNN) has proven to be highly efficient. We employed the idea of a "convolution", a sliding window or "filter" that passes through the image, defining and evaluating important features one at a time, then reducing them to their essential features and repeating the process. Authors in [12] suggested CNN text classification techniques. Fig. 5 demonstrates a text-processing CNN architecture. It starts with an input sentence split into embedding words or words: low-dimensional representations created by models such as word2vec or GloVe. Words are divided into characteristics and fed into a convolutionary layer. The convolution results are either "pooled" or aggregated to a representative amount. This number is fed to a neural network that is completely connected, making a classification decision based on the weights assigned to each function within the text.

- Hierarchical Attention Network (HAN): HAN has been presented in [13] and is based on the same concept of the Attention GRU. HAN architecture is built using bi-directional GRU to acquire the word context. It contains two levels of attention for both word and sentence. Word2vec model is used to construct the required word embeddings. As shown in Fig. 6, HAN consists of several layers: word-encoder-layer, word-attention-layer, sentence-encoder-layer, sentence-attention-layer and fully-connected-layer.

- Recurrent Convolution Neural Network (RCNN): RCNN architecture is a combination of RNN and CNN to use the advantages of both techniques in a model. The authors in [14] proposed a text classification method based on RCNN as shown in Fig. 7. The main idea of this technique is capturing the required useful data by recurrent network and constructing the text representation using the convolutional structure.

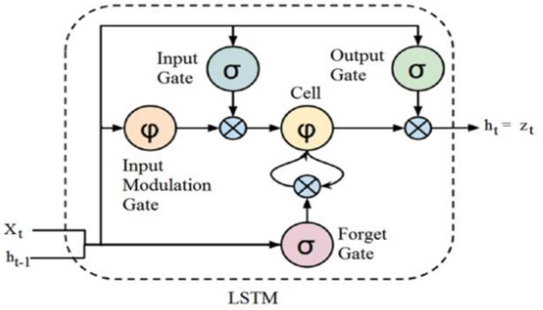

Fig. 3. LSTM Cell Architecture.

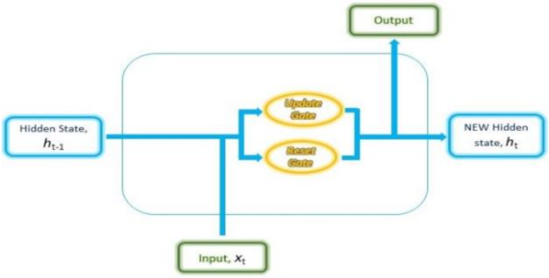

Fig. 4. GRU Cell Architecture.

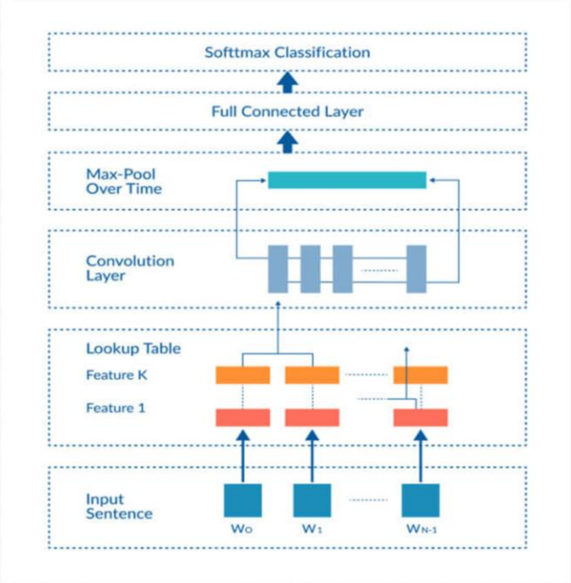

Fig. 5. CNN Architecture for Text Classification [12].

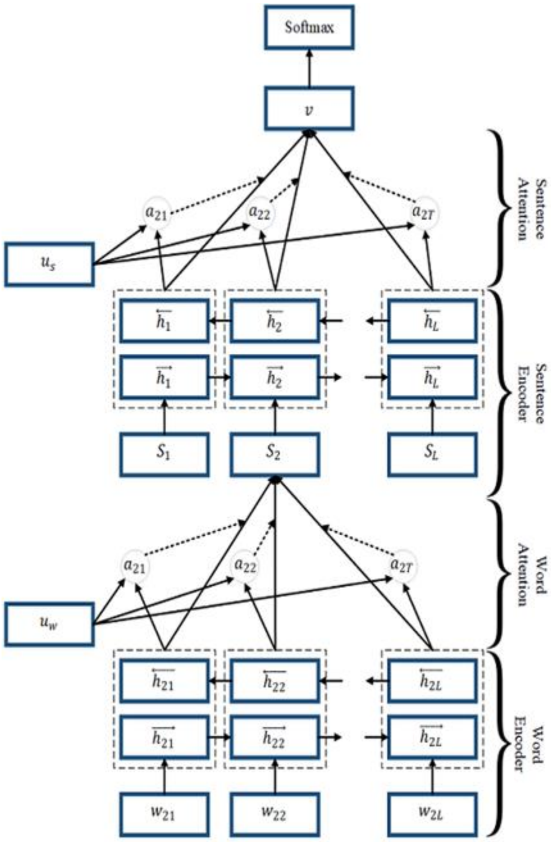

Fig. 6. HAN Architecture [13].

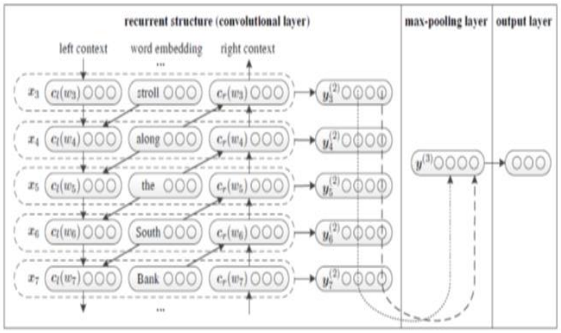
recurrent structure (convolutional layer)
max-pooling layer
output layer
left context
word embedding
right context
x_3 $c_l(w_3)$ stroll $c_r(w_3)$ $y_3^{(2)}$
x_4 $c_l(w_4)$ along $c_r(w_4)$ $y_4^{(2)}$
x_5 $c_l(w_5)$ the $c_r(w_5)$ $y_5^{(2)}$
x_6 $c_l(w_6)$ South $c_r(w_6)$ $y_6^{(2)}$
x_7 $c_l(w_7)$ Bank $c_r(w_7)$ $y_7^{(2)}$
$y^{(3)}$

Fig. 7. RCNN Architecture for Text Classification [14].

- Random Multimodel Deep Learning (RMDL): Deep learning models across many areas have produced state-of-the-art tests. As shown in Fig. 8, architecture for classification of Random Multimodel Deep Learning (RDML) resolves the issue of getting the perfect design and structure from deep learning classifiers. RDMLs have the ability of accepting variety of input data including text, image, pictures and symbols. RMDL contains 3 random units, one left DNN classifier, one middle Deep CNN classifier, and one right Deep RNN classifier.

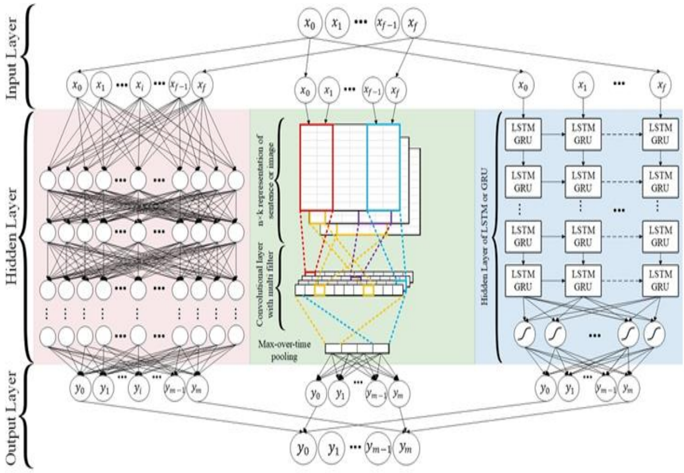
Input Layer
Hidden Layer
Output Layer
x_0 x_1 x_{f-1} x_f
x_0 x_1 x_i x_{f-1} x_f
n×k representation of sentence or image
Convolutional layer with multi filter
Max-over-time pooling
Hidden Layer of LSTM or GRU
LSTM GRU
y_0 y_1 y_i y_{m-1} y_m
y_0 y_1 y_{m-1} y_m

Fig. 8. RMDL Architecture [15].

IV. Experiments and Results

In this section, comprehensive results of the proposed methodologies will be discussed. The experiments were tested on the dataset available on UCI repositories for the collection of SMS spam. This dataset benchmark includes a total of 5574 English-language emails. Non-spam and spam numbers were respectively 4827 and 747[16]. Metrics like accuracy, precision, recall and F measure are used to assess the proposed methodologies. Table I shows the confusion matrix of Non-Spam and Spam SMS.

The various metrics are measured via the following equations:

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$F = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (4)$$

All classical machine learning experiments described in Section III have been conducted using Auto Model; Auto Model is an expansion of RapidMiner Studio; it speeds up the model construction and validation process. This produces a mechanism that can be altered or created. This helps analyze the data given, provides appropriate models for problem solving, and helps to contrast the obtained results. All deep learning tests are performed using Keras and the python deep learning package. For all the deep learning experiments, the messages are converted into semantic word vectors using Glove model [17]. All networks begin with embedding layer that contains the semantic vectors resulted from Glove. Furthermore, Dropout layers are added to all deep learning models described in Section III to decrease the complexity of connections among the fully connected dense layers. The model of DNN contains 6 dense layers and 5 dropout layers. The model of LSTM contains 4 LSTM layers and 3 dropout layers in addition to the start embedding and final dense layers. The same structure is built for GRU model; it contains 4 GRU layers and 3 dropout layers. The architecture of CNN contains

7 convolution layers, 7 max pooling layers and 3 dropout layers. The architecture of HAN model has the same layers described in Section III [13]. RCNN model contains 4 convolution layers, 4 max pooling, 4 LSTM and 1 dropout layer. RDML model contains 3 DNNs, 3 CNNs, 3 RNNs as explained in Section III [15].

The findings of the proposed methods are presented in Table II and Fig. 9. Results showed that, relative to classical machine learning techniques, deep learning architectures have made significant improvements. Machine learning classifiers' highest accuracy is 96.86% according to Gradient Boosted trees. The highest accuracy resulted from RDML is 99.26 % of all possible methodologies.

TABLE. I. NON-SPAM AND SPAM CONFUSION MATRIX

	Non-Spam	**Spam**
Non-Spam	True-Positive (TP)	False-Positive (FP)
Spam	False-Negative (FN)	True-Negative (TN)

TABLE. II. THE RESULTS OF ALL PROPOSED METHODS

Proposed Classifier	**Accuracy**	**Precision**	**Recall**	**F Measure**
Naive Bayes	86.75	86.74	100.0	92.90
GLM	93.78	93.78	99.42	96.52
Fast Large Margin	95.10	95.04	99.49	97.21
Decision Tree	94.91	94.83	99.57	97.14
Random Forest	96.73	96.77	99.56	98.14
Gradient Boosted Trees	96.86	96.98	99.49	98.22
SVM	86.68	86.69	100.0	92.86
DNN	93.84	92.39	99.93	96.01
LSTM	97.88	97.82	99.56	98.68
GRU	98.12	98.23	99.44	98.83
CNN	98.24	98.25	99.57	98.90
HAN	99.17	99.09	99.65	99.37
RCNN	99.05	99.05	99.62	99.33
RMDL	**99.26**	99.25	99.59	99.42

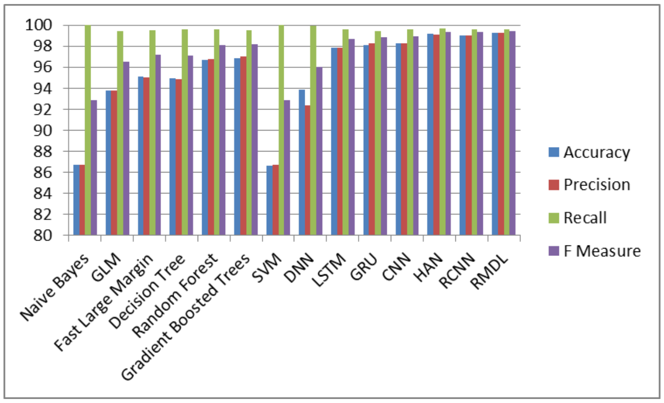

Fig. 9. Comparison of all the Proposed Methods.

TABLE. III. COMPARISON OF RMDL AND PREVIOUS DEEP LEARNING ARCHITECTURES

Classifier	**Accuracy %**
RDML	99.26
3CNN [8]	**99.44**
SLSTM [9]	99.01
CNN [10]	98.4
CNN [11]	99.1

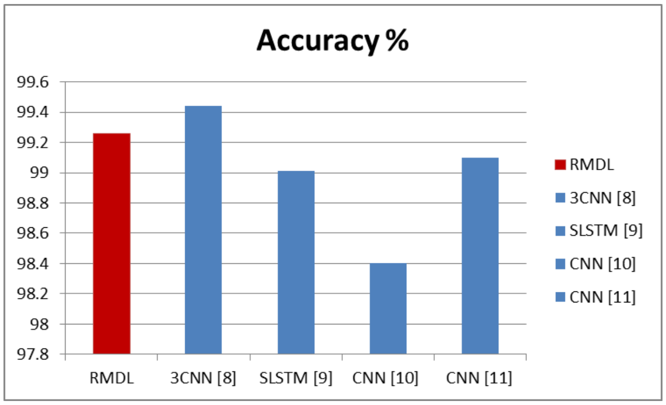

Fig. 10. Comparison of RDML and Previous DL Algorithms.

The main advantage of RMDL architecture is the power of finding the appropriate architecture for deep learning while at the same time reducing error rates and improving accuracy. The RMDL models include the advantages of the architecture of DNN, CNN and RNN. Table 3 and Fig. 10 provide a contrast between RMDL's accuracy and the best accuracy in the four deep learning articles [8-11] listed in the related work section. Using complicated 3CNN architecture in [8], the best accuracy was achieved.

V. CONCLUSION AND FUTURE WORK

This study covers SMS spam filtering task; thirteen proposed machine learning and deep learning classifiers were evaluated. Six classical machine learning classifiers were tested: Decision Tree, Naïve Bayes, Fast Large Margin, GLM, Random Forest, Gradient Boosted tress and SVM. Seven deep learning architectures were evaluated: DNN, LSTM, GRU, CNN, RCNN, HAN and RMDL. Benchmark dataset that contains 5574 messages was evaluated. The best accuracy of machine learning classifiers is 96.86 % resulted from Gradient Boosted trees. The best accuracy of all proposed methodologies is 99.26 resulted from RDML. These results indicate the power of using deep learning architectures on text classification modules. Our future work will focus on two main stages; the first stage is testing the model of Hierarchical Deep Learning for Text (HDLTex) [18]. The second stage is applying the techniques of transfer learning [19] that achieved promising accuracies on different classification tasks. of RMDL architecture is the power of finding the appropriate architecture for deep learning while at the same time reducing error rates and improving accuracy. The RMDL models include.

REFERENCES

[1] Bayomi-Alli, O., Misra, S., Abayomi-Alli, A., &Odusami, M. (2019). A review of soft techniques for SMS spam classification: Methods, approaches and applications. Engineering Applications of Artificial Intelligence, 86, 197-212.

[2] Shafi'I, M. A., Latiff, M. S. A., Chiroma, H., Osho, O., Abdul-Salaam, G., Abubakar, A. I., &Herawan, T. (2017). A review on mobile SMS spam filtering techniques. IEEE Access, 5, 15650-15666.

[3] Lota, L.N., Hossain, B.M., 2017. A systematic literature review on SMS spam detection techniques. Int. J. Inf. Technol. Comput. Sci. 7, 42–50.

[4] Chaudhari, N., Jayvala, V., &Vinitashah, P. (2016). Survey on Spam SMS filtering using Data mining Techniques.International Journal of Advanced Research in Computer and Communication Engineering, 5(11).

[5] Kazi, K. F. I., &Dharmadhikari, S. C. (2014). Preserving the value of SMS texting: A survey on mobile SMS spam classification techniques and algorithms. Data Mining and Knowledge Engineering, 6(3), 106-112.

[6] Sajedi, H., Parast, G. Z., &Akbari, F. (2016). Sms spam filtering using machine learning techniques: A survey. Machine Learning Research, 1(1), 1-14.

[7] Delany, S. J., Buckley, M., & Greene, D. (2012). SMS spam filtering: methods and data. Expert Systems with Applications, 39(10), 9899-9908.

[8] Roy, P. K., Singh, J. P., & Banerjee, S. (2020). Deep learning to filter SMS Spam. Future Generation Computer Systems, 102, 524-533.

[9] Jain, G., Sharma, M., &Agarwal, B. (2019). Optimizing semantic LSTM for spam detection. International Journal of Information Technology, 11(2), 239-250.

[10] Popovac, M., Karanovic, M., Sladojevic, S., Arsenovic, M., &Anderla, A. (2018, November). Convolutional Neural Network Based SMS Spam Detection. In 2018 26th Telecommunications Forum (TELFOR) (pp. 1-4).IEEE.

[11] Gupta, M., Bakliwal, A., Agarwal, S., &Mehndiratta, P. (2018, August). A Comparative Study of Spam SMS Detection Using Machine Learning Classifiers. In 2018 Eleventh International Conference on Contemporary Computing (IC3) (pp. 1-7). IEEE.

[12] Kim, Y. (2014). Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.

[13] Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., &Hovy, E. (2016, June). Hierarchical attention networks for document classification. In Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies (pp. 1480-1489).

[14] Lai, S., Xu, L., Liu, K., & Zhao, J. (2015, February). Recurrent convolutional neural networks for text classification.In Twenty-ninth AAAI conference on artificial intelligence.

[15] Kowsari, K., Heidarysafa, M., Brown, D. E., Meimandi, K. J., & Barnes, L. E. (2018, April). Rmdl: Random multimodel deep learning for classification. In Proceedings of the 2nd International Conference on Information System and Data Mining (pp. 19-28).ACM.

[16] Almeida, T. A., Hidalgo, J. M. G., &Yamakami, A. (2011, September). Contributions to the study of SMS spam filtering: new collection and results. In Proceedings of the 11th ACM symposium on Document engineering (pp. 259-262).ACM.

[17] Pennington, J., Socher, R., & Manning, C. (2014, October). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

[18] Kowsari, K., Brown, D. E., Heidarysafa, M., Meimandi, K. J., Gerber, M. S., & Barnes, L. E. (2017, December). Hdltex: Hierarchical deep learning for text classification. In 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA) (pp. 364-371).IEEE.

[19] Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification.arXiv preprint arXiv:1801.06146.