ISSN 2278-3091
Volume 9, No.5, September - October 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse326952020.pdf
https://doi.org/10.30534/ijatcse/2020/326952020

Deep SMS Spam Detection using H2O Platform

Dima Suleiman[1,*], Ghazi Al-Naymat[2,3], Mariam Itriq[1],
[1]Information Technology Department, The University of Jordan, Amman, Jordan
Dima.suleiman@ju.edu.jo
[2] Department of IT, College of Engineering and Information Technology, Ajman University, UAE
[3] Computer Science Department, King Hussein Faculty of Computing Sciences, Princess Sumaya University for Technology, Amman, Jordan

ABSTRACT

SMS (Short Message Service) is a mean of communication between people which can either be spam or ham. SMS spam is a major concern since it is annoying and many people do not like to receive it. In this paper, deep learning and random forest machine learning algorithms are used to determine the most important features that can be used as input to classifiers. In deep learning, as the number of neurons, and epochs increase the precision, recall, F-measure, and accuracy will increase which is good, however the runtime will increase and this will affect the efficiency negatively so that the best number of neurons and epochs that can achieve high precision, recall, F-measure and accuracy and at the same time low runtime is 20 for the number of neurons and 10 for epochs. In a random forest, also the runtime will increase as the number of trees and the maximum depth of each tree increase. In order to achieve the best values for precision, recall, F-measure, and accuracy using random forest, the number of trees must be 5 and the value of maximum depth of each tree must not exceed. The main contribution of the research is proposing a new SMS detection classifier that is based on using H2O platform. Comparisons are made between several machine learning algorithms which are deep learning, random forest and naïve bays. In all experiments, the evaluation model that was used is the cross-validation with 3-folds and 10-folds cross-validations. The experiments are conducted on SMS messages dataset that was proposed by UCI Machine Learning Repositories. The experimental results show that a naïve bay was the best in terms of runtime with a value of 0.6 seconds while it was the worst in terms of accuracy performance. On the other hand, the random forest provides the best performance with values 96%, 86%, 91%, and 0.977% for precision, recall, F-measure, and accuracy, respectively where 50 trees are used with a maximum depth equal to 20.

Key words: SMS spam; Random Forest; Naïve Bays; Deep Learning; H2O.

1. INTRODUCTION

Spam is an abbreviation of the English concept (Sending and Posting Advertisement in Mass). This concept was associated with the beginnings of the spread of the e-mail service via the Internet (Spam e-mails), but it soon became associated with the short message service (SMS) provided by smartphone devices. Due to the increase in using the short messages, in addition to its low cost compared to other communication services that need an Internet connection, a challenge called (Spam SMS) appeared. SMS spam messages include all unwanted messages such as marketing advertisements and promotions messages that contain rumors and free services [1].

Email and SMS spams are annoying and cause service performance degradation [2]. SMS spam usually reachesa cluster of individuals and broadcasted through a network of mobile. However, the World-Wide net transfers email spam.

Several solutions of SMS spam detection were exported from successful email anti-spam solutions [3,4], but not all ways of email spam solutions are applicable and suitable for SMS spam [5]. The reasons for this inconsistency refer to the fact that: the size of the SMS message is small, SMS free of Multi-purpose Internet Mail Exchanger (MIME) and SMS supports only textual representation. Several techniques were utilized for SMS spam detection, such as using artificial neural networks, K-nearest neighbors (KNN), support vector machines (SVM), naive Bayes (NB), decision trees, random forests, and hybrid methods [6]. The outcomes of the experiments indicated that the SVM and NB classifiers provided the highest accuracy [7], but the techniques such as logistic regression, decision trees and Bayesian classification were still time-consuming [7]. Most of existing studies use Weka in their experiments [8,9].

In this paper, a new SMS spam detection methodology is proposed which is an extension of the previous work [10]. Comparisons are made between deep learning (DL), random forest (RF), and naïve bays (NB) machine learning algorithms. The matrics that are used for evaluating the models are the accuracy, precision, recall, F-measure, and runtime.

All experiments are performed over the H2O platform [11]. In addition to finding the most effective classifier, tuning of deep learning and random forest parameters are made in order to find the best values that will achieve high performance. The dataset that is used for the experiment is the same dataset that was proposed in UCI Machine Learning Repositories.

The remainder of this paper is organized as follows: Section 2 addresses the related work. The proposed framework is

* Corresponding author. Tel.: +962-6-5359949; fax: +962-6-5347295.
E-mail address: d.suleiman@psut.edu.jo

explained in section 3. Section 4 discusses the experimental results. Finally, the conclusion is presented in section 5.

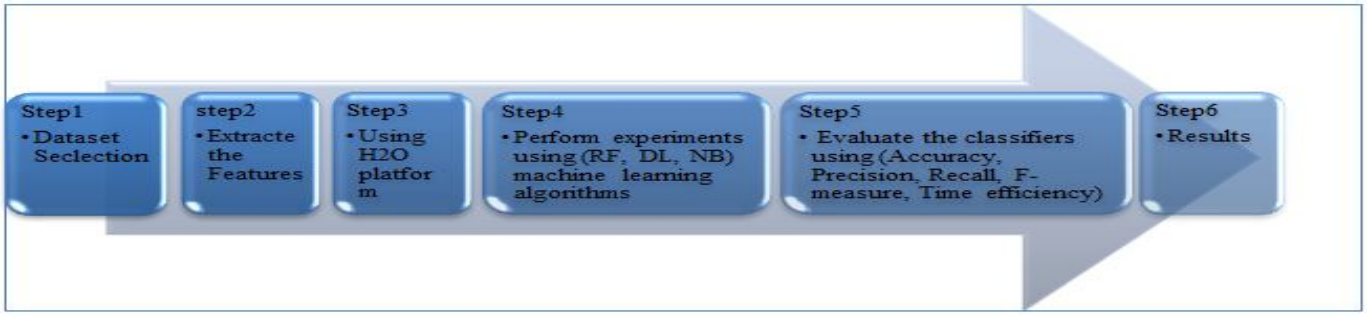


Figure 1: Steps of Proposed Method

2. RELATED WORKS

Unlike email, SMS spam detection technique has some challenges due to the SMS features such as the restricted message size, limited header information, there is no real database for SMS spam, Moreover, the frequent use of abbreviations and acronyms in SMS messages, increase the extent of ambiguity that cannot be detected by traditional email spam detection schemes. Furthermore, using the same email spam detection classifies for classifying SMS spams, needs massive memory storage and processing resources which are not available in the smartphones environment [8][12].

Due to the importance of this topic, considerable work had been done and new detection methods were proposed. [13] presented a review of the currently available methods, challenges, and future research directions on spam filtering and detection techniques. As a result of their review, they found that the support vector machine and the Bayesian network can be used as SMS spam classifiers. At the same time, the review showed that many effective bio-inspired algorithms such as Monkey Search, Cat Swarm, Magneto Tactic Bacteria Optimization depend on Moment Migration, Chicken Swarm Optimization, the Bat algorithm, the Cuckoo search algorithm, the Bees algorithms, and Particle Swarm Optimization were not able to be used as SMS spam detection classifiers. [14] presented a new machine learning classification algorithm to detect and filter the spam SMS based on 10 features(Presence of mathematical symbols, mobile number, URLs, dots, special symbol, emotions, Lowercased words, Uppercased words, Keyword specific and Message length) Also, they used 5 machine learning algorithms namely Decision Table, Logistic
Regression, Naïve Bayes,J48, and Random Forest. Out of these algorithms, the one that achieved the best results(96.1% true positive rate) was the Random Forest Classification Algorithm. [9]proposed another spam filtering technique using different machine learning algorithms and the experiments shows that TF-IDF with Random Forest classification algorithm outperformed other algorithms in terms of accuracy with a value of (79.50%). On the other hand, the Support Vector Machines (SVM) algorithm [15] provided better performance than other classifiers such as Naïve Bayes, Multinominal Naïve Bayes, and KNearest Neighbor with the different number of K= 1,3 and 5 where the average accuracy was 98.9%. Table1 shows the comparisons between different SMS spam detection classifiers in the period between 2007 and 2016.

3. PROPOSED FRAMEWORK

The proposed SMS spam detection framework consists of the following step: The first step is the selection of the dataset. After that, the features of the text are extracted followed by choosing the appropriate machine learning platform, in this research, we used the H2O platform. At the next step, we used several machine learning algorithms: random forest (RF), deep learning (DL), and naïve Bayes (NB).

The experiments are conducted on the dataset where the results are evaluated using Accuracy, Precision, Recall, F-measure, Time efficiency evaluation measures. The steps of the proposed framework can be seen in Figure 1.

3.1 Data Selection

The experiments are performed on the UCI Machine Learning repository dataset which was gathered in 2012 [24]. The SMS messages are classified into either spam or ham messages. The total number of SMS messages is 5574 where 747 messages are spam while the others are ham. The messages in the dataset are saved in a text file. Each row in the file represents a text and the type of the message if it is spam or ham.

Table 1: Comparisons between different SMS spam detection classifiers

Ref#	Dataset	Pre-processing/ Features	Methods/ classifier	Evaluation	Results
[16] 2007	1) English(1002 ham, 82 spam) 2) Spanish (1157 ham, 199spam)	Bigram and trigram of characters and words bigrams,Lowercase words	Bogofilter DMC LR OSBF-Lua SVM	10-fold cross Validation, Using ROC curve	1) Performance of SVM was good. 2) Differences between filters not clear and need a larger dataset.
[5] 2011	UCI Machine Learning repository	Tokenization(two types)	SVM, EM classifier, MDL and C4.5 and others	Accuracy (Acc%). Spam Caught (SC%), Blocked Hams (BH%), and Matthews CorrelationCoefficient (MCC)	1) Accuracy of using SVM was 97.5%. 2) 50% of spams are correctly filtered
[17] 2013	UCI Machine Learning repository	NB, Laplace smoothing, tokenization	SVM KNN Ensemble(Random forest, Adaboost	70% training set and 30% testing, 10-fold cross-validation	spams caught (SC) was 93%, the Error rate was 1.5% and blocked hams (BH) was 74% A reduced error by more than half
[18] 2013	http://www.dt.fee.unicamp.br/~tiago/smsspamcollection 4827 SMS ham messages and 747 spam messages	Tokenization(two types)	SVM, Logical regression, Expectation-Maximization (EM) clustering	Blocked Hams , Spam Caught, Accuracy , Matthews Correlation Coefficient	Accuracy was 97%
[19] 2013	Turkish(430 ham, 420 spam) English (425 spam, 450 ham)	TF-IDF,BoW, SF, number of significant terms,upper or lower case characters, numeric and alphanumeric characters such as (“!”, “$”), in addition to URL link.	SVM, KNN	chi-square, Gini index, (GI) metrics	
[20] 2014	6600 messages (Volunteers)	The bigrams frequency, monograms, message size	The artificial neural network, Decision tree and Naïve Bayes	Accuracy	Accuracy 93%
[21] 2014	http://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection 4827 SMS ham messages and 747 spam	Words semantic, capital words, spam words, SMS Segments, SMS Frequency, Unique Words, URL, Using word “Call”, The URL rate, linguistic processes, psychological processes and Spoken features	SVM, Boosting, Random Forest SVM is the best	Precision,recall, F-measure, accuracy, ROC curve	Accuracy 92%-98%
[22] 2015	Tiago’s dataset Consists of 5572 messages (4827 ham, 747 spam)	Tokenization, Stop words . Symbols such as “The”, “☺” will be removed	GentleBoost	Accuracy	Accuracy 98%
[23] 2016	Dataset contains 400 message, 200 spam and 200 ham	discard words will be removed	Naïve Bayes classifier and Apriori Algorithm	Accuracy, Precision, Recall	

3.2 Feature extraction and selection

The performance of the SMS spam detection classifiers is highly affected by the feature extraction process. While some features may increase the quality of the SMS detection classifier, other features may decrease the quality. From the related work, different features are used and summarized in Table 2. In order to get the features, most of the methods made tokenization to split the message into tokens. Tokenization process is important in order to get the tokens, then the frequency of each token will be calculated for further analysis, however, some methods make stemming and stop words removal [22], while others think that it is not good to make such processing [5][17]. The most important features are:

- Message Length (ML): The message total number of characters. Most of research used the message length as an indication of message class [19][20][25]:

Message length must not exceed 160 characters, and then the spammer such as marketing spammers will try to use the maximum length, so that spam messages are longer than the ham messages [5].

- Number of message words (NW): in most cases ham messages have less number of words [25].

- Total number of words that contain less than three charaters(NW3) normalized by the maximum number of words less than three.

- Ratio of the number of words with length less than three characters (NW3) according to the total number of words (NW) [25] as in equation (1).

$$RNW3 = \frac{NW3}{NW} \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (1)$$

Usually, ham messages contain short words, such as shortcuts.

- Total number of Capital (CW) normalized by the maximum number of words

- Ratio of the number of Capital (CW) according to the total number of words (NW) [21] as in equation (2).

$$RCW = \frac{CW}{NW} \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (2)$$

The existence of capital letter words in most cases is an indication of spam messages.

Table 2: Preprocessing in previous research

Ref#	Preprocessing
[16]	o Words o Lowercased words o Bi-grams and Tri-grams Character o Bi-grams words
[19]	o Length of the message o Number of terms o Characters that are uppercase o Characters that are non-alphanumeric o Characters that are numeric o Existence of URL
[20]	o Length of the message o Matching spam words o Match found in the list of di-grams
[21]	o Words that are capital letters o Segments of SMS o Words that are unique o Existence of URL o Existence of word “Call o URL rate
[25]	o Length of the message o Characters that are alphanumeric o Characters that are numeric o Each letter frequency o Special chars frequency (10 chars: *, _ ,+,=,%,$,@, , \,/) o Number of words o Number of short words less than three letters o Number of characters in a word o Average length of the word. o Average length of sentence in chars. o Average length of sentence in words o Punctuation chars frequency:. ‹; ? ! : ( ) – “ « » <> [ ] { }

- Total number of alphanumeric characters (AC) normalized by the maximum number of alphanumeric characters.
- Ratio of alphanumeric characters (AC) according message length (ML) [25] as in equation (3).

$$RAC = \frac{AC}{ML} \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (3)$$

- Total number of special characters (SC) such as “*, _ ,+,=,%,$,@, , \,/” normalized by maximum number of special characters.

- Ratio of special characters (SC) such as “*, _ ,+,=,%,$,@, , \,/” according message length (ML) [19][25] as in equation (4).

$$RSC = \frac{SC}{ML} \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (4)$$

- Total number of punctuation characters (PC) such as “‹; ? ! : ( ) – “ « » <> [ ] { }” normalized by maximum number of punctuation characters.

- Ratio of punctuation characters (PC) such as "ʿ; ? ! : ( ) – " « » <> [ ] { }" according message length (ML) [19][25] as in equation (5).

$$\text{RPC} = \frac{PC}{ML} \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (5)$$

- Total number of digit characters (DC) which are normalized by the maximum number of digit characters.

- Ratio of digit characters (DC) according message length (ML) [19][25] as in equation (6).

$$\text{RDC} = \frac{DC}{ML} \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (6)$$

SMS spam messages in most cases have special, punctuation and digit characters more than SMS ham messages

- The existence of the word "Call" [20][21], however in this research we take into consideration the existence of digits in addition to the word "call" in the same message (Callcount).

In most researches they used the word "call" as criteria for detecting spam messages, however, we find that the existence of the word "call" alone without digits is used a lot in ham messages, such as saying "call me" to your friend, however, using the word "call" with number represent the phone number is an indication of spam.

- The existence of URL [19][21] in the message is an indication that the message probability of being spam is high.

URL in SMS message is evidence for the existence of spam in the message.
The total number of features is sixteen, but not all features are important for SMS spam detection. We used the H2O platform to determine the most important features using deep learning and random forest. We found that the most important features after using deep learning were the existence of URL then the existence of the word "call" and digits as shown in Figure 2. On the other hand, the most important features after using random forest were the total number of digits and the ratio of digits as shown in Figure 3.
In this research, we selected the most common and important features, for example instead of using the total number of digits and the ratio of digits we will choose one of them. The selection will depend on the most important one which can be determined according to the results of applying deep learning and random forest in SMS messages.

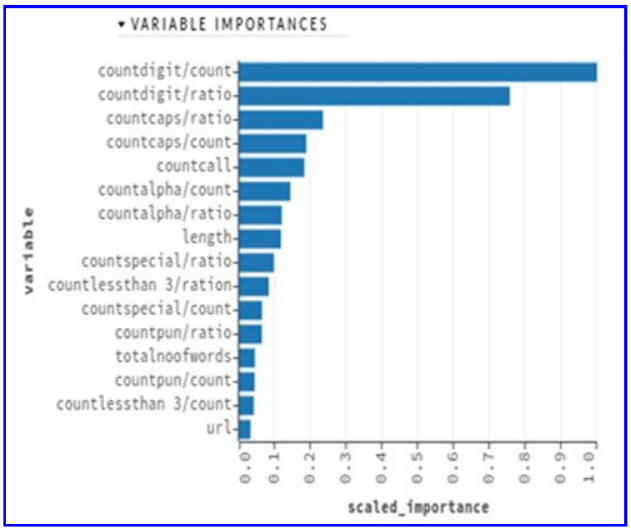


Figure 2: Important Features using Deep Learning

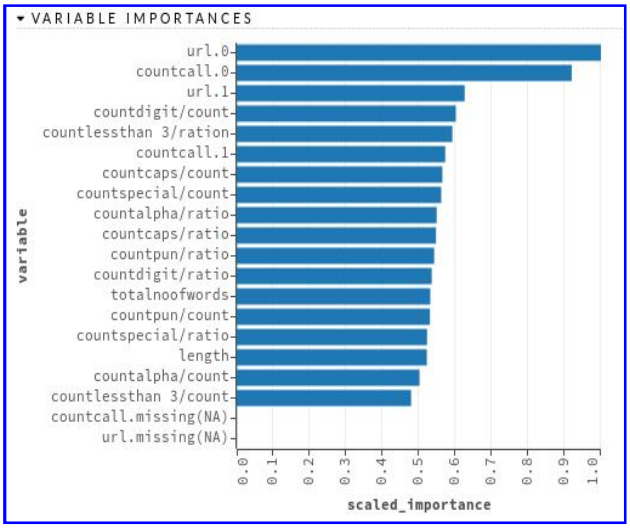


Figure 3: Important Features using Random Forest

3.3 Evaluation metrics

Five metrics were used to evaluate the quality of the SMS detection classifiers which are: Precision, recall, accuracy, F-measure, and runtime. The confusion matrix shown in Table 3 was used for evaluating the binary class classification. True positive (TP) are messages that are correctly classified as spam. False positive (FP), the SMS messages that are ham but they are classified as spam. False negative (FN) are the messages that are classified as ham while they are spam. Finally, true negative (TN) are the messages that are correctly classified as ham. Therefore, based on the previous definitions, the recall, accuracy, F-measure, and runtime can be calculated as follows [26][27]:

• Accuracy: the percentage of the messages that are classified correctly over the total number of messages as in equation (7)

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (7)$$

• Precision (P): how many of classified messages are spam message in equation (8)

$$\text{Precision} = \frac{TP}{TP + FP} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (8)$$

• Recall (R): how many of the spam messages are correctly classified as spam as in equation (9)

$$\text{Recall} = \frac{TP}{TP + FN} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (9)$$

• F-measure: this measure combines two metrics which are precision and recall into one measure, as in equation (10)

$$F - measure = \frac{2 * P * R}{P + R} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (10)$$

In order to get better classification, F-measure and accuracy values must be high. On the other hand, the runtime value must be low. Therefore, we have to make a balance between the five measures.

Table 3: Confusion Matrix

Actual label	Predicted label	
	Spam	Ham
Spam	True positive (TP)	False negative (FN)
Ham	False positive (FP)	True negative (TN)

4. EXPERIMENTAL RESULTS

Three classifiers are used in our experiments: RF, DL, and NB. The experiments in this research are divided into two parts: the first part was to make tuning for selected algorithms and the second part is to make comparisons between the classifiers in order to select the best one. However, in all experiments, five runs for each change are made, and the average is taken. The two parts will be covered in the following subsections.

4.1 Tuning of Classifiers

1) Deep Learning

Deep learning has many parameters; we select three of them for tuning which are: Activation Function, Number of hidden neurons, and epochs. The first parameter to tune is the activation function, there are many activation functions that can be used in deep learning such as: tanh, tanhwithdropout, rectifier, rectifierwithdropout, maxout, maxoutwithdropout. In SMS spam detection the best activation function was the rectifier. The second parameter to tune is the number of neurons in hidden layers, the default value for it is 200 neurons, however after applying the default value which is 200 neurons the results give good precision but at the same time the runtime is high, and in this model number "200" is high so that we tried values 5, 10,20,100. As the number of neurons increases, accuracy, precision, recall, and F-measure will increase also the runtime will increase which is not desired, so that we tried to find the value that will keep accuracy, precision, recall, and F-measure high and at the same time the runtime low the best value was 20 neurons for each hidden layer. The last parameter is the epochs, as the number of epochs increases all metrics will increase, the accuracy, precision, recall, and F-measure in addition to runtime. Increasing of accuracy, precision, recall, and F-measure is needed however, increase in runtime is not good since we need spam detector to be fast so that the best value of epochs that keep the runtime reasonable and the same time keep the accuracy, precision, recall, and F-measure high is 10. Table 4 shows the experimental results after tuning the parameter, each row in the table is the average of five runs.

2) Random forest

In order to improve the performance of using random forest; parameters are tuned such as the number of trees and the maximum depth of each tree, according to experiments shown in Table 5. The optimal values that achieve the best results are 5 trees and 20 maximum depth. During the experiments, the increase in maximum depth does not show a significant improvement in accuracy, precision, recall, and F-measure but instead results in increasing of the runtime which is not desired; so that the author tries to find the best value for maximum depth that improve accuracy, precision, recall, and F-measure in addition to runtime.

4.2 Comparisons between classifiers

In this research, we used 3-fold and 10-fold cross-validation models to evaluate the classifiers. In 3-fold cross-validation, we divided the dataset into three parts where two of them were used in training and one was used in testing. In 10-fold cross-validation instead of dividing the dataset into three parts as in 3-fold, the dataset was divided into 10 parts. In 10-fold cross-validation, 9 parts were used in training and one part was used for testing. In order to achieve reliability, 3-fold experiments were repeated 3 times while in 10-fold cross-validation, the experiments were repeated 10 times. Finally, the average values of all experiments were taken. The results showed that 10-fold cross-validation provided better results in terms of accuracy, precision, recall, and F-measure but it was slower. The three classifiers that are used are DL, RF and NV.

The best results of using DL were achieved when the rectifier activation function was used; the number of epoch is 10 and 20 neurons used in each hidden layer. On the other hand, the best results of using RF were achieved when the number of epochs is 10, the number of neurons in the hidden layer is 20, the number of trees is 5 and the maximum depth is 20. All the results of using the three classifiers with 3-fold cross-validation are shown in Table 6, and the result of using 10-fold cross-validation is shown in Table 7.

Table 4: The values of accuracy, precision, recall, and runtime of deep learning algorithm after tuning the parameters

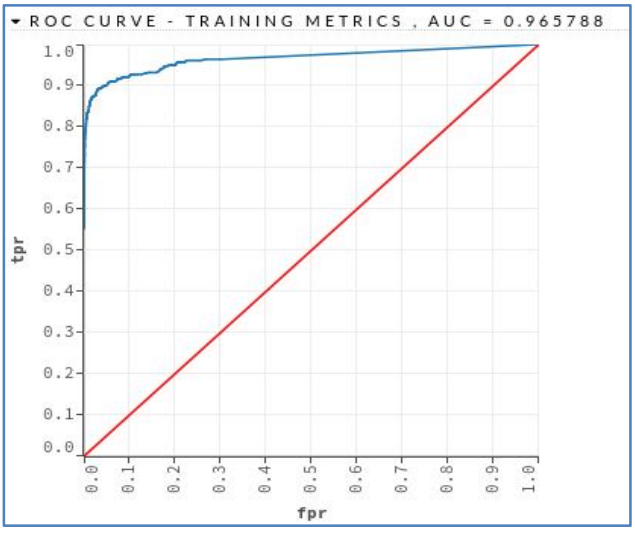
Nfold	Hidden	Epoch	Activation function	Run Time	F-measure	Accuracy	Precision	Recall	RMSE
3	5,5	10	Rectifier	0.003795116	0.847170614	0.961626	0.908373	0.794551	0.185613
3	10,10	10	Rectifier	0.005559953	0.856589794	0.964304	0.926839	0.796849	0.178198
10	10,10	10	Rectifier	0.015343079	0.867272588	0.966557	0.926907	0.816771	0.176289
10	20,20	10	Rectifier	0.028739577	0.87178574	0.968178	0.943079	0.812477	0.170604
10	20,20	50	Rectifier	0.109675753	0.882303902	0.969147	0.939424	0.825366	0.167626
10	100,100	10	Rectifier	0.270718298	0.877150788	0.968966	0.938604	0.825147	0.171647
10	100,100	50	Rectifier	1	0.877922548	0.97039	0.94588	0.827948	0.166198

Table 5: The values of accuracy, precision, recall, and runtime of the random forest algorithm after tuning the parameters

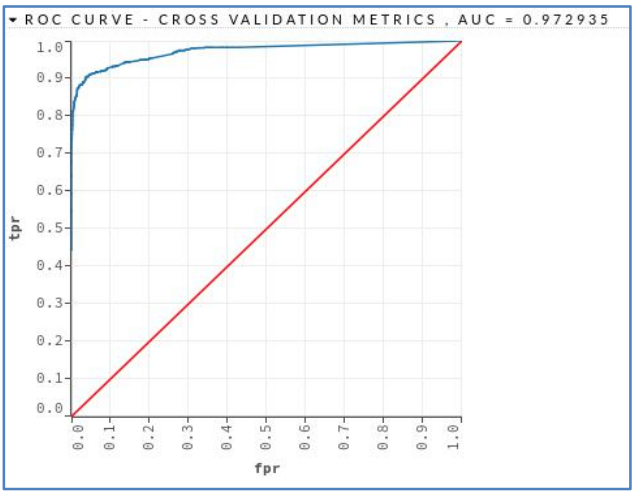
Nfolds	Ntrees	max_depth	Run Time	Accuracy	F-measure	Precision	Recall	RMSE
3	5	20	0.045607662	0.967704	0.873233	0.923219	0.830067	0.172003
3	5	5	0.068824306	0.964231	0.854987	0.933127	0.789817	0.180399
10	5	20	0.120871863	0.971528	0.888239	0.936218	0.846865	0.163988
3	10	5	0.145409511	0.964519	0.857359	0.929667	0.79611	0.179401
3	10	10	0.201486129	0.969304	0.877911	0.941032	0.824025	0.165218
3	15	5	0.206836196	0.965548	0.861061	0.936518	0.797514	0.178023
3	20	5	0.2443z52708	0.966104	0.86229	0.939973	0.796886	0.178022
10	10	20	0.247192867	0.974049	0.898624	0.943221	0.859389	0.157181
3	15	15	0.282298547	0.972475	0.89024	0.95499	0.834129	0.160357
3	15	10	0.286955086	0.970769	0.883314	0.950931	0.825817	0.163753
3	30	5	0.294649934	0.965671	0.86163	0.936489	0.798463	0.178281
3	30	30	0.318229855	0.973246	0.893701	0.954347	0.840705	0.158183
3	20	20	0.343989432	0.972421	0.891755	0.941572	0.847559	0.16045
3	40	20	0.37655218	0.973419	0.894788	0.952594	0.844487	0.157459
3	50	30	0.424009247	0.974576	0.899793	0.950617	0.854802	0.154813
3	50	20	0.431208719	0.97455	0.899656	0.954105	0.851645	0.156316
10	20	20	0.521466314	0.976307	0.905575	0.957359	0.860661	0.15296
10	30	20	0.724042272	0.97704	0.90994	0.957867	0.868004	0.151447
10	50	20	1	0.976826	0.908676	0.959669	0.864444	0.150712

The experiment showed that, in DL algorithm, if the number of epochs and the number of neurons in each layer are increased, the accuracy, precision, recall, and F-measure will also increase. However, unfortunately, the runtime will increase. Therefore, we have to make a balance between accuracy, precision, recall, and F-measure and runtime. The best values for the parameter to achieve just a balance are to use 10 epochs and to have up to a maximum of 20 neurons in each layer. In RF, if the number of trees and the depth of each tree increased, then the accuracy, precision, recall, F-measure, and the runtime will increase. The best results are achieved by using 5 trees and the maximum depth of the tree is 20

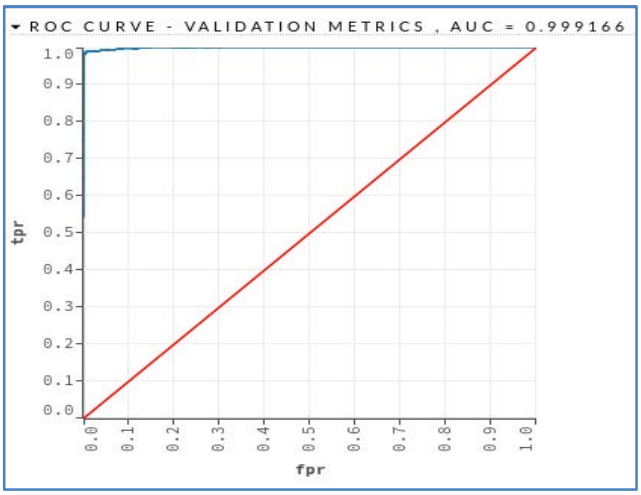
The results of Table 7 shows that RF is the best classifier in terms of accuracy, precision, recall, and F-measure where the values are 0.97%95%, 85% 0.89% respectively. On the other hand, the best classifier in terms of the runtime is NB where the value is 0.6 seconds. In order to get the best values with respect to runtime in addition to precision, recall, F-measure, and accuracy, tuning must be made in order to make balance. The chart for ROC distribution can be shown in Figure 4, the results are for random forest.



a) ROC Curve for Training Metrics



b) ROC Curve for Cross Validation Metrics



c) ROC Curve for Validation Metrics

Figure 4: ROC Distributions

Table 6: Comparisons between RF, DL, NB algorithms in terms of A is Accuracy, F is F-measure, P is Precision and R is Recallusing 3-fold cross-validation

Algorithm		Run Time	A	F	P	R
RF	Ntrees = 15, Max_depth =15	0.020	0.972	0.890	0.955	0.834
RF	Ntrees = 30, Max_depth =30	0.023	0.973	0.894	0.954	0.841
RF	Ntrees = 50, Max_depth =20	0.031	0.975	0.900	0.951	0.851
RF	Ntrees = 40, Max_depth =20	0.027	0.973	0.895	0.953	0.845
RF	Ntrees = 15, Max_depth =10	0.021	0.971	0.883	0.951	0.826
RF	Ntrees = 50, Max_depth =30	0.031	0.975	0.899	0.951	0.855
RF	Ntrees = 20, Max_depth =20	0.025	0.972	0.892	0.942	0.848
RF	Ntrees = 10, Max_depth =10	0.015	0.964	0.878	0.941	0.824
RF	Ntrees = 20, Max_depth =5	0.018	0.966	0.863	0.940	0.797
RF	Ntrees = 15, Max_depth =5	0.015	0.966	0.861	0.937	0.798
RF	Ntrees = 30, Max_depth =5	0.021	0.966	0.862	0.936	0.798
RF	Ntrees = 5 , Max_depth =5	0.005	0.964	0.855	0.933	0.790
RF	Ntrees = 10, Max_depth =5	0.011	0.965	0.857	0.930	0.796
DL	Hidden =10,10 , Epoch = 10	0.006	0.964	0.857	0.927	0.797
RF	Ntrees = 5 , Max_depth =20	0.003	0.968	0.873	0.923	0.830
NB		0.001	0.960	0.840	0.920	0.773
DL	Hidden =5,5 , Epoch = 10	0.003	0.962	0.847	0.908	0.795

5. CONCLUSION

SMS is one of the most important communication media between people. There are two types of SMS messages which are: spam and ham. Spam messages are the undesirable messages that the user does not want to receive and must be removed or blocked before that while the ham messages are the desirable messages. Many of SMS spam detection classifiers are already exist, but there is no guarantee that they are 100% efficient. Most of the previous research used Weka for making experiments, however, this work used the H2O platform in order to improve the performance of SMS spam classifiers. The experiments were made in the UCI Machine Learning Repositories dataset. This research contains more than one contribution; the first one is the use of machine learning algorithms that are already built-in H2O for determining the most important features. The machine learning algorithms that were used are deep learning and random forest and the most important features are the number of digits and the existence of URL in the message text. The second contribution was the tuning of deep learning and random forest algorithms in order to get efficient SMS spam detector in terms of precision, recall, F-measure, accuracy, and runtime. The parameters that tuned in deep learning are the activation function, the number of neurons in the hidden layer, and epochs, the results showed that the best values of these parameters are the rectifier activation function, 20 neurons for each hidden layer, and 10 epochs. Also tuning was made in random forest parameters for the number of trees and maximum depth for each tree. The best values are 5 trees and a maximum depth of 20. The last contribution was in proposing a new classifier for SMS spam detection that over H2O as a platform. The three classifiers were used which are:

DL, RF, and NB. Two validation models are used including 3-fold and 10-fold cross-validation. Each experiment is repeated 5 times and the average of the run was taken. In a conclusion, 10-fold cross-validation provides better results in terms of accuracy, precision, recall, and F-measure while 3-fold experiments were the best in terms of runtime. The experimental results showed that the random forest classifier was the best in terms of accuracy, precision, recall, and F-measure where the values were 0.977%, 96%, 86%, and 91% respectively while NB was the worst. On the other hand, the NB classifier was the best in terms of runtime.

Table 7: Comparisons between RF, DL, NB algorithms in terms of A is Accuracy, F is F-measure, P is Precision and R is Recallusing 10-fold cross-validation

Algorithm		Run Time	A	F	P	R
RF	Ntrees = 50, Max_depth =20	0.072	0.977	0.909	0.960	0.864
RF	Ntrees = 30, Max_depth =20	0.052	0.977	0.910	0.958	0.868
RF	Ntrees = 20, Max_depth =20	0.038	0.976	0.906	0.957	0.861
DL	Hidden =100,100 , Epoch = 50	1	0.970	0.878	0.946	0.828
RF	Ntrees = 10, Max_depth =20	0.018	0.974	0.899	0.943	0.860
DL	Hidden =20,20 , Epoch = 10	0.029	0.969	0.872	0.943	0.812
DL	Hidden =20,20 , Epoch = 50	0.110	0.970	0.882	0.940	0.825
DL	Hidden =100,100 , Epoch = 10	0.271	0.969	0.877	0.939	0.825
RF	Ntrees = 5, Max_depth =20	0.009	0.972	0.888	0.936	0.847
DL	Hidden =10,10 , Epoch = 5	0.015	0.967	0.867	0.927	0.817
NB		0.003	0.960	0.836	0.910	0.775

REFERENCES

1. Mohammadi, A. and Hamidi, H., "**Analysis and evaluation of privacy protection behavior and information disclosure concerns in online social networks**, " International Journal of Engineering, Transactions B: Applications, Vol. 31, No. 8, (2018), 1234-1239.
2. G. Cormack. "**Email Spam Filtering: A Systematic Review**, "Foundations and Trends in InformationRetrieval, 1(4):335–455, 2008.
3. El-Alfy, E.-S.M. and AlHasan, A.A., "**Spam filtering framework for multimodal mobile communication based on dendritic cell algorithm, Future Generation Computer Systems**, " Vol. 64, (2016), 98-107.
4. H. Ji and H. Zhang, "**Analysis on the content features and their correlation of web pages for spam detection**, "Communications, China, vol. 12, pp. 84-94, 2015.
5. Tiago A. Almeida , José María G. Hidalgo , AkeboYamakami, "**Contributions to the study of SMS spam filtering: new collection and results**, " Proceedings of the 11th ACM symposium on Document engineering, September 19-22, 2011, Mountain View, California, USA [doi:10.1145/2034691.2034742].
6. Sajedi H., Parast G., Akbari F., "**SMS Spam Filtering Using Machine Learning Techniques:A Survey**, " Machine Learning Research 2016; 1(1): 1-14, doi: 10.11648/j.mlr.20160101.11.
7. Chaudhari N., Jayvala, Vinitashah, "**Survey on Spam SMS filtering using Data mining Techniques**, " International Journal of Advanced Research in Computer and Communication Engineering ISO 3297:2007 Certified Vol. 5, Issue 11, November 2016.
8. T. Mahmoud, A. Mahfouz, "**SMS Spam Filtering Technique Based on Artificial Immune System**, " IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 2, No 1, March 2012 ISSN (Online): 1694-0814.
9. N. N. Amir Sjarif, N. F. MohdAzmi, S. Chuprat, H. M. Sarkan, Y. Yahya, and S. M. Sam, "**SMS Spam Message Detection using Term Frequency-Inverse Document Frequency and Random Forest Algorithm**," Procedia Computer Science, vol. 161, pp. 509–515, 2019, doi: 10.1016/j.procs.2019.11.150
10. D. Suleiman and G. Naymat, "**Sms spam detection using h2o framework**, "Procedia Computer Science, Vol. 113, (2017), 154-161, doi:10.1016/ j.procs.2017.08.335.
11. D. Suleiman, M. Al-Zewairi, and G. Naymat, "**An Empirical Evaluation of Intelligent Machine Learning Algorithms under Big Data Processing Systems**,"Procedia Computer Science, vol. 113, pp. 539–544, 2017, doi: 10.1016/j.procs.2017.08.270.
12. D.-N. Sohn, J.-T. Lee, and H.-C. Rim, "**The contribution of stylistic information to content-based mobile spam fltering**, " in Proceedings of the ACL-IJCNLP Conference Short Papers., 2009.
13. Abdulhamid, S.M.; AbdLatiff, M.S.; Chiroma, H.; Osho, O.; Abdul-Salaam, G.; Abubakar, A.I.; Herawan, T. "**A Review on Mobile SMS Spam Filtering Techniques**, " IEEE Access 2017, 5, 15650–15666.
14. Choudhary, N., & Jain, A. K.."**Towards filtering of SMS spam messages using machine learning based technique**, " In International Conference on Advanced Informatics for Computing Research (pp. 18-30). Springer, Singapore. (2017, March).

15. NilamNur Amir Sjarif, YazriwatiYahya, SuriayatiChuprat, Nurul Huda Firdaus Mohd Azmi.“**Support Vector Machine Algorithm for SMS Spam Classification in The Telecommunication Industry**, ”International Journal on Advanced Science, Engineering and Information Technology, Vol. 10 (2020) No. 2].
16. Gordon V. Cormack , José María Gómez Hidalgo , Enrique PuertasSánz, “**Feature engineering for mobile (SMS) spam filtering,**” Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, July 23-27, 2007, Amsterdam, The Netherlands [doi;10.1145/1277741.1277951]..
17. Shirani-Mehr, H. (2013). “**SMS spam detection using machine learning approach**, ” CS229 Project 2013, Stanford University, USA, pp. 1–4.
18. T. Almeida, J. M. G. Hidalgo, and T. P. Silva. “**Towards sms spam filtering: Results under a new dataset**, ”International Journal of Information Security Science, 2(1), 2013.
19. Uysal AK, Gunal S, Ergin S, .“**The impact of feature extraction and selection on SMS spam filtering**, ” Electronics and Electrical Engineering 2013; 19(5): 67–72.
20. Mujtaba, G., Yasin, M., “**SMS spam detection using simple message content features**, ” J. Basic Appl. Sci. Res. 4, 275–279 (2014).
21. A. Karami and L. Zhou, “**Improving static SMS spam detection by using new content-based features**, ” 20th Americas Conference on Information Systems, AMCIS 2014..
22. Akbari, F., Sajedi, H.“**SMS spam detection using selected text features and boosting classifiers**. ” In: Information and Knowledge Technology. IEEE (2015).
23. SABLE S, “**SMS CLASSIFICATION BASED ON NAIVE BAYES CLASSIFIER AND SEMI-SUPERVISED LEARNING**, ” INTERNATIONAL JOURNAL OF INNOVATIONS IN ENGINEERING RESEARCH AND TECHNOLOGY , VOLUME 3, ISSUE 7, July-2016.
24. SMS Spam Collection Data Set from UCI Machine Learning Repository, http://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection.
25. Anchal, Sharma A., “**SMS Spam Detection Using Neural Network Classifier**, ” International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 6, June 2014.
26. Chinchor N and Sundheim B. “**MUC-5 evaluation metrics**, ” In: Proceedings of the 5th conference on message understanding, Baltimore, MD, 25 August 1993, pp. 69–78. Stroudsburg, PA: Association for Computational Linguistics.
27. Wang D, Navathe SB, Liu L et al. “**Click traffic analysis of short URL spam on Twitter**, ” In: Proceedings of the 9th international conference on collaborative computing: networking, applications and worksharing (collaboratecom), Austin, Texas, USA, 20 October 2013, pp. 250–259. New York: IEEE.