# Identifying spam SMS using Apache Spark MLlib

[1]Atanu Ghosh, [2] Mr. Ajit Kumar Pasayat
[1]Department of Computer Science and Engineering, Centurion University of Technology and Management, Bhubaneswar, Odisha, India.
[2]Assistant Professor, Department of Computer Science and Engineering, Centurion University of Technology and Management, Bhubaneswar, Odisha, India.

***Abstract :*** Short Message Service (SMS) has grown huge now days because of its flexibility and making communication more effortless to the mobile phone users. At the same time, increasing popularity of SMS and reduction of the cost of messaging service has made advertiser more interesting to send unsolicited commercial advertisements (spam) to users. Spam messages are annoying and many people do not want to get. There are many methods available to detect spam messages. Different classifiers which depend on Naïve Bays, Support Vector Machine and many other ML algorithms were already used. In this paper we compared different classifiers which mainly depend on Apache Spark MLlib library to evaluate accuracy and runtime. Here we used Logistic Regression with L-BFGS (Limited-memory Broyden–Fletcher–Goldfarb–Shanno), Naïve Bays, Decision Tree and Gradient Boosted Trees to compare the vectors. Besides using different classifiers, also descried the most important features that were being used as input to Decision Tree, Gradient Boosted Trees, Naïve Bays and Logistic Regression with L-BFGS classifiers. Features those are helps to detect spam SMS mostly the existence of URL and the number of digits present in a SMS. The experiment used the dataset which is proposed by UCI Machine Learning Repositories. For this experiment the dataset was split into two parts so that 80% of the data were taken as training purpose and 20% of the data were taken as testing purpose. Therefore, experiments show that Naïve Bays is the faster algorithm to achieve best accuracy than others. It took 3.16 seconds to achieve 95% accuracy on test data.

***Keywords— SMS Spam, Naïve Bays, Decision Tree, Logistic Regression with LBFGS, Gradient Boosted Trees, Apache Spark MLlib***

## I. INTRODUCTION

Short Message Service (SMS) has gained very much popularity nowadays because it can be used as a substitute for voice call. Situations like where voice call not possible or may be undesirable, SMS played an important role. Also growth of mobile phone user day by day makes SMS more popular way of communication. Communications through standardized protocols used to transmit short text messages between mobile phone users. The thriving vogue of text messaging service makes telecom operator bound to provide service at cheaper price. This led us seen a surge in number of unsolicited commercial advertisement sent to message inbox by the advertiser. SMS spam or mobile phone spam is any junk that delivered to mobile phone as text message. Spam SMS are not only annoying but also cause ruin phone memory. In some countries even receiver also charged for receiving SMS. So it is obligate to prevent SMS spam as soon as it received at the mobile station, if not before that. There are many techniques which have been applied to Identify spam SMS. We can categorize them into two ways: one is non content based approach which is used by telecom operators and another is content based approach which is used by mobile phone user. Text classification is one of the content based approach that can be used as classifying messages either spam or ham. Ham messages are those which are created by noble users while spam messages are created by promotional companies. But for identifying spam SMS there is a problem that may yield when ham messages are misclassified as spam [1].

SMS spam-filtering and email spam-filtering are different in a number of ways. The main limitation for SMS spam-filtering is the scarcity of available real dataset whereas large variety of dataset present for email spam-filtering. Moreover, small length and informal language of text messages makes spam-filtering much complex than email spam-filtering.

There are different kind of techniques for SMS spam identification, such as deep learning, support vector machine(SVM), Naïve Bayes(NB), random forest, k-nearest neighbor, decision tree(DT), in addition to hybrid methods [2]. Nevertheless, different analogies and experiments were drawn up for spam identification using various techniques and various datasets. The results appeared such that NB and SVM produced maximum accuracy [3] whereas techniques like logistic regression, decision tree and Bayesian classification undergo with time consuming [3] problem.

In this paper, the authors introduce a new SMS spam identification method that mainly based on Apache™ Spark MLlib as platform. There are many research papers available using Weka for SMS spam classification [4], [5] but Apache™ Spark MLlib is completely new in this race. Various machine learning algorithms such as logistic regression with L-BFGS, NB, DT and gradient boosted trees will be used for comparison. Accuracy and runtime of different algorithms will be calculated using the dataset available in UCI Machine Learning Repositories. Beside this, PySpark of Apache™ Spark will be integrated in Java integrated development environment IntelliJ IDEA for exploring dataset and processing.

The reminder of this paper will be formed as follows: section 2 will describe about related theories section 3 will overview which will comprise information about Apache™ Spark MLlib and classification algorithms that are being used. Section 4 will be about proposed method which will include the detail about selected dataset as well as feature extraction, method that proposed by the authors. experimental results will be shown in section 5. Finally, conclusion and future scope of this work will be explained in section 6.

## II. RELATED THEORIES

There are large number of researches available for email spam detection (e.g. [6], [7], [8]), but for SMS spam identification a few number of studies available in literature. Cormak et. al. [9] endeavored to get better result for SMS spam filtering by applying the filters (e.g. Bogofilter, Logistic regression, Dynamic Markov Compression, SVM and OSBF) that used in email spam classification. But after exploring the results, it was educed that all evaluation filters were not clearly differentiate.

Email spam detection classifiers are not able to perform well for SMS spam identification because text messages have limited features, lake of real dataset, concise length of text and the informal language of writing text. Hence, a real dataset was created in 2011 UCI Machine Learning Repository and made available publicly [1]. In addition, Tiago A. Almeida et. al. [1] used two tokenizers in their work. One is for splitting words in patterns which comprises commas, dots and colons at the center like email and another tokenizer split the series of characters separated by blanks, dots, commas and others. Shirani-Mehr in his paper [10], did not use pre-processing stemming but special characters were withdrawn and dataset was split into tokens.

The accuracy and the performance of different classifiers depend upon feature extraction and selection from text messages of the dataset. Uysa AK et. al. in their paper [11], explore the impact of feature extraction and selection using two datasets. One is English which consist of 425 spam and 450 ham messages. Another one is Turkish, consists of 420 spam and 430 ham messages. Term Frequency and Inverse Document Frequency(TF-IDF) were applied to calculate frequency term and to illustrate the document as collection of words and their frequencies, vector space model was exercised.

## III. OVERVIEW

### Apache™ Spark framework

Apache Spark MLlib is a scalable machine learning library from Apache™ foundation. It is an open source framework which helps to work with different machine learning algorithms. MLlib supports different languages like R, Python, Scala, Java and can be able to manipulate any Hadoop data source. Apache™ spark is easy to deploy therefore, anyone having existing Hadoop clusters can easily run spark and MLlib. Besides machine learning algorithm it also provides distributed linear algebra, statistical algorithms and several more features. For more details, please refer to Spark documentation for machine learning library guide [12].

### Classification models

Structured or unstructured data can be classified using classification algorithms in machine learning. Classification is a technique for categorizing input data and recognizes the class under new data will fall under. Here we used four classification algorithms for text classification and those are: Logistic regression with L-BFGS, Naïve Bayes, Decision tree and Gradient boosted trees.

***a) Logistic regression with L-BFGS****:* This method is broadly used to predict binary response. It is a linear statistical method for exploring dataset that have one or more independent variables which infer an outcome. The equation of logistic loss is:

$$L(w; x, y) := \log(1 + \exp(-yw^T x)) \quad (1)\ [13]$$

When new data point x will be given, logistic regression model makes prediction using logistic function:

$$f(z) = \frac{1}{1+e^{-z}} \quad (2)\ [13]$$

Where $z = w^T x$. and w is a weight vector. If $f(w^T x) > 0.5$, the outcome is by default positive otherwise negative.

For faster convergence the authors used Logistic regression with limited memory BFGS. Constructing and inverting Hessian matrix explicitly is not possible for multiclass problems. L-BFGS use only an approximation to true Hessian and the simple form the enabled simple analytic inversion of the Hessian is use to build the approximation iteratively up.

***b) Naïve Bayes****:* This method is a simple multiclass classification algorithm with the assumption of independence between every pair of features [14]. So in a simple illustration, Naïve Bayes classifier comprise that the availability of an exact feature in a class is not related to the availability of any other feature. All the properties contribute to the probability even though the features rely on each other or upon the existence of other feature. It calculates the conditional probability distribution of every feature conferred label and thereafter applying Bayes' theorem it calculates the conditional probability distribution of label conferred an observation and manage it for prediction.

***c) Decision tree:*** This is also a popular method for classification and regression in machine learning. The algorithm is extensively used because of handling categorical features, enhanced to multiclass classification setting, no need of feature scaling and also it can capable to accept non-linearity as well as feature interactions. It breaks down the dataset and developed an association decision tree in a similar time period. The decision tree is also a greedy based algorithm, use to execute recursive binary partitioning of feature space. Taking the best split from a set of possible splits, each and every partition is chosen greedily for maximizing the information gain at tree node.

Let assume that s is a split that partitions the dataset D whose size is N. Let, two partitions are made and those are $D_{left}$ and $D_{right}$ of size $N_{left}$ and $N_{right}$ respectively. Then the information gain $IG(D, s)$ is:

$$IG(D,s) = Impurity(D) - \frac{N_{left}}{N} Impurity(D_{left}) - \frac{N_{right}}{N} Impurity(D_{right}) \quad (3)\ [15]$$

Here, *Impurity* is a measure of the homogeneity of the labels at a node. There two *Impurity* provided by the current implementation of Spark MLlib for classification. Those are Gini impurity and Entropy impurity.

***d) Gradient-Boosted trees*:** Gradient boosted trees(GBTs) generate predictive models in the form of an ensemble of decision tree. It trains the model in a stage wise fashion to minimize a loss function. GBTs manage categorical features, enhanced to the multiclass setting similar like decision tree does. There is no need for feature scaling and also it can able to accept non-linearity as well as feature interactions. Every iteration the GBTs classifier exercise present ensemble for predicting the label of every training

instance and then compare the result with actual label. The method runWithValidation helps to eliminate overfitting problem for GBTs by Validating data while training [16].

## IV. PROPOSED METHOD

In this paper, the authors proposed a new method for identification of SMS spam. A sequence of steps was proposed for identifying spams, those are: selection of real dataset using which the classifiers will be trained, after that selection and extraction of features from the dataset will performed. Using Apache™ Spark as a platform different classifiers will be trained. Here authors used four classification algorithms: Linear regression with L-BFGS, Naïve Bayes, Decision trees and Gradient boosted trees. Next accuracy and time efficiency of each and every algorithm will be calculated and finally evaluation of experimental results will be made in order to compare between classifiers. The figure 1 shows complete process for spam identification.

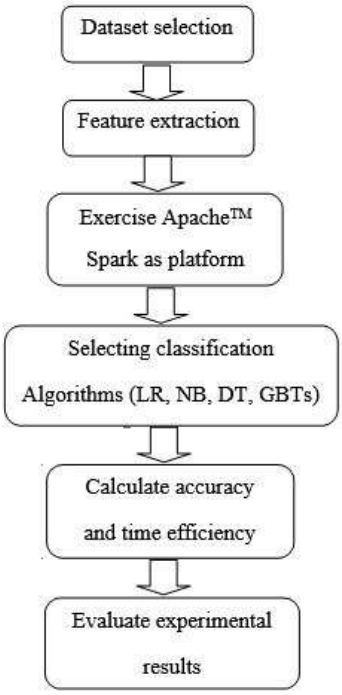


Fig. 1. Proposed method for spam identification.

### *A. Dataset Selection*

Dataset that present in UCI Machine Learning Repository [17] was selected by authors. The dataset was donated to repository in 2012. 5574 text messages are present in the dataset and categorized into two types, spam and ham. 747 numbers of spam messages and 4827 numbers of ham messages are introduced in the dataset. Among spam messages, 425 messages were gathered from Grumbletext online forums and 322 messages were gathered from Corpus v.0.1 Big. On the other hand, among ham messages, 3375 messages were collected indiscriminately from NUS SMS Corpus and 450 messages were excerpted from PhD thesis for Caroline Tag. Remaining 1002 ham messages were collected from Corpus v.0.1 Big. The dataset is saved as text format and have labeled with spam and ham for spam messages and ham messages respectively. Table 1 shows some examples from dataset.

TABLE I. EXAMPLE OF DATASET

| | |
|---|---|
| ham | Go until jurong point, crazy.. Available … |
| ham | Ok lar... Joking wif u oni... |
| spam | Free entry in 2 a wkly comp to win FA Cup final tkts 21st… |
| ham | U dun say so early hor... U c already then say... |
| ham | Nah I don't think he goes to usf, he lives around here though |
| spam | FreeMsg Hey there darling it's been 3 week's now and no… |

### *B. Feature extraction*

Extracting features from dataset is one of the most important step to be considered. The performance of spam identification will be affected by the feature extraction. For extracting feature, the authors used vectorization method that provided by Apache™ Spark. Term frequency-inverse document frequency (TF-IDF) [18] is used for extracting feature from text. If $f_{ij}$ is the *frequency* of term $i$ in document $j$ then *term frequency* $TF_{ij}$ will be

$$TF_{ij} = \frac{f_{ij}}{max_k f_{kj}} \quad (4)\ [18]$$

Again if $i$ term appears in $n_i$ of the $N$ documents in the collection then inverse document frequency will be

$$IDF_i = log_2(N/n_i) \quad (5)\ [18]$$

The term TF.IDF will calculated by the product of TF and IDF as

$$TF.IDF = TF_{ij} \times IDF_i \quad (6)\ [18]$$

## V. EXPERIMENTAL RESULTS

After Here we use four machine learning algorithms logistic regression with L-BFGS, Naïve Bayes, decision tree and gradient boosted trees for text classification. All of the four algorithms validate on the dataset that present in UCI Machine Learning Repository. Although, we made different alternation of parameters of each classifier and took the average of four runs of each alternation. Figure 2 shows the comparison of time efficiency and accuracy of each algorithm. It shows that Naïve Bayes algorithm is the best fit for spam classification among the four classifiers that we have chosen. Naïve Bayes produced 95% of

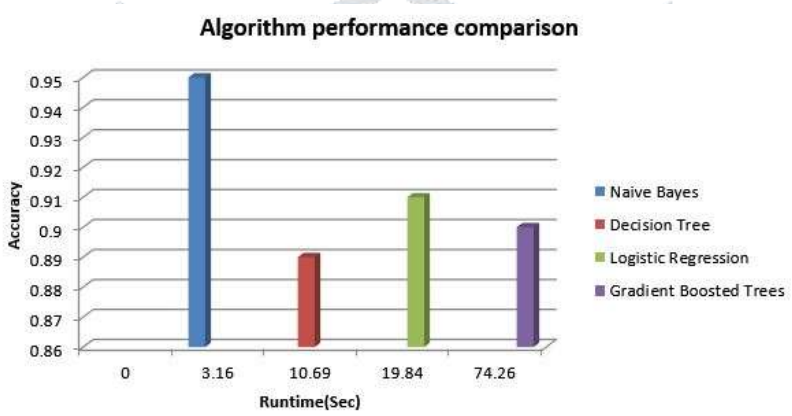


Fig. 2. Performance comparison between NB, DT, LR and GBTs.

accuracy in 3.16 second of training. The performance of these algorithm somehow based on configuration of the system and the system with higher configuration will produce results more swiftly.

Here authors used Apache™ Spark as a platform which provide MLlib machine learning library and no other research paper on SMS spam identification is available using this library. MLlib has a collection of machine learning algorithms like LR, DT, NB, GBTs and many more. All the works are made on IntelliJ IDEA which integrated with py4j of PySpark (The Spark Python API).

## VI. CONCLUSION AND FUTURE SCOPE

Now a days SMS is one of the most momentous way of communication. Popularity of this service makes advertiser more enchanting to send promotional advertisement. Spam messages are so much annoying and not desirable. Several comparisons have done for eliminating spam from ham messages. In this paper, authors compared performance between four algorithm LR, NB, DT and GBTs to evaluate model. Moreover, authors used Apache™ Spark as platform to compute efficiency between algorithms. Several research papers [4], [5] are available using same dataset but the environment and the classifiers that have been taken are completely different. Accuracy and time efficiency have been considered for performance comparison. The results defined that GBTs took larger time to classify spam messages while NB is better in accuracy and runtime. For better performance we could use Hadoop distributed file system(HDFS) in future.

## REFERENCES

**[1]** Tiago A. Almeida, José María G. Hidalgo, Akebo Yamakami, 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.

**[2]** Sajedi H., Parast G., Akbari F., SMS Spam Filtering Using Machine Learning Techniques:A Survey, Machine Learning Research 2016; 1(1): 1-14http://www.sciencepublishinggroup.com/j/mlr doi:10.11648/j.mlr.20160101.11

**[3]** 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.

**[4]** Zainal, K., Sulaiman, N. F., Jali, M. Z.: An Analysis of Various Algorithms for Text Spam Classification and Clustering Using RapidMiner and Weka. International Journal of Computer Science and Information Security (IJCSIS), 13(3), pp. 66–74. 2015.

**[5]** Kawade D., Oza K., SMS Spam Classification using WEKA, International Journal of Electronics Communication and Computer Technology.

**[6]** G. Cormack, "Email Spam Filtering: A Systematic Review," Foundations and Trends in Information Retrieval, vol. 1, no. 4, pp. 335–455, 2008.
**[7]** T. A. Almeida, A. Yamakami, and J. Almeida, "Evaluation of Approaches for Dimensionality Reduction Applied with Naive Bayes Anti-Spam Filters," in Proceedings of the 8th IEEE International Conference on Machine Learning and Applications, Miami, FL, USA, 2009, pp. 517–522.
**[8]** J. M. Gómez Hidalgo, "Evaluating Cost-Sensitive Unsolicited Bulk Email Categorization," in Proceedings of the 17th ACM Symposium on Applied Computing, Madrid, Spain, 2002, pp. 615–620.
**[9]** Gordon V. Cormack , José María Gómez Hidalgo , Enrique Puertas Sánz, 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
**[10]** Shirani-Mehr, H. (2013). SMS spam detection using machine learning approach. CS229 Project 2013, Stanford University, USA, pp. 1–4
**[11]** 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
**[12]** Machine Learning Library (MLlib) guide for Apache spark[online], https://spark.apache.org/docs/latest/ml-guide.html
**[13]** Linear Methods - RDD-based API for Apache Spark [Online], https://spark.apache.org/docs/latest/mllib-linear-methods.html
**[14]** Naive Bayes - RDD-based API for Apache spark [Online], https://spark.apache.org/docs/2.2.0/mllib-naive-bayes.html
**[15]** Decision Trees - RDD-based API for Apache Spark [Online], https://spark.apache.org/docs/2.2.0/mllib-decision-tree.html
**[16]** Ensembles - RDD-based API for Apache Spark [Online], https://spark.apache.org/docs/latest/mllib-ensembles.html
**[17]** SMS Spam Collection Data Set from UCI Machine Learning Repository [Online], http://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection
**[18]** Rajaraman, A.; Ullman, J.D. (2011). "Data Mining". Mining of Massive Datasets (PDF). pp. 1–17. doi:10.1017/CBO9781139058452.002. ISBN 978-1-139-05845-2.