Cost-Sensitive Classifier for Spam Detection on News Media Twitter Accounts (revised April 2017)

Georvic Tur
Department of Computer Science
and Information Technology,
Simón Bolívar University, Valle Sartenejas,
Baruta, Edo. Miranda
Apartado 89000, Caracas, Venezuela.
Email: 12-11402@usb.ve

Masun Nabhan Homsi
Department of Computer Science
and Information Technology,
Simón Bolívar University, Valle Sartenejas,
Baruta, Edo. Miranda
Apartado 89000, Caracas, Venezuela.
E-mail: mnabhan@usb.ve.

***Abstract*—Social media are increasingly being used as sources in mainstream news coverage. However, since news is so rapidly updating it is very easy to fall into the trap of believing everything as truth. Spam content usually refers to the information that goes viral and skews users views on subjects. Despite recent advances in spam analysis methods, it is still a challenging task to extract accurate and useful information from tweets. This paper aims at introducing a new approach for classification of spam and non-spam tweets using Cost-Sensitive Classifier that includes Random Forest. The approach consisted of three phases: preprocessing, classification and evaluation. In the preprocessing phase, tweets were first annotated manually and then four different sets of features were extracted from them. In the classification phase, four machine learning algorithms were first cross-validated aiming at determining the best base classifier for spam detection. Then, class imbalanced problem was dealt by resampling and incorporating arbitrary misclassification costs into the learning process. In the evaluation phase, the trained algorithm was tested with unseen tweets. Experimental results showed that the proposed approach helped mitigate overfitting and reduced classification error by achieving an overall accuracy of 89.14% in training and 76.82% in testing.**

***Keywords*—Spam Classification, Twitter, Topic Discovering, Cost-Sensitive Classifier, Random Forest**

I. Introduction

Since the very moment that the first computer network was created, spamming became a possibility. Indeed, this is the only prerequisite for it to exist. It does not matter what type of network it is being used on, because what spam looks like and how it works depend on its environment [1]. However, in its essence, spam always refers to an instance of repetitious behavior. At first, the word Spam was only the name of a food item in Britain. Then it was used as a joke that relied on repetition. By the time the first computer networks were implemented, this word found a new usage, because spam was everything repetitive, inattentive, and vexing. This was the first era of spam, and it lasted until 1995 [1]. From that year until 2003, the full implications of this phenomenon were already recognized. While malicious users had recognized its monetary potential, governments were attempting to stop it, since its victims could be afflicted bandwidth problems and even financial loss [1]. From 2003 onwards, as several massive social networks were born, spamming became even more specialized, and so the techniques used for stopping it. Now it is very common to use machine learning (ML) for spam detection [1]. Even when the problem is circumscribed to Twitter, previous studies have found different specialized ways of detecting it by identifying the most important features according to their criteria and using them with well-known ML algorithms. Providing the methods for detecting patterns automatically on vast amounts of gratuitous data has become a staple of the contemporary machine learning field. With those patterns, one can then attempt to make predictions [2]. Usually input data consists of several features that are thought to be relevant to a given problem. When at least one of the given features is the target class, then the model that trains on them uses a supervised learning approach [2]. On the other hand, if the target class is not available for training, then the model uses an unsupervised learning approach [2]. Additionally, there is third kind of learning approach used in machine learning and data mining: reinforcement learning. This is the case whenever simple agents that follow punish-and-reward strategies are used to solve a given problem [2]. In this paper, we used a supervised learning approach because we provided a dataset of tweets with a labeling for training. Five algorithms that use supervised learning were used in this research. They are Naive Bayes (NB), Adaboost (AB), K-Nearest Neighborhood (KNN), Random Forest (RF) and Cost-Sensitive Classifier (CSC). NB algorithm is a classification algorithm whose most important characteristic is its assumption of independence among the attributes present in the dataset. The maximum posterior probability principle is used to select the best value for the class [3]. AB is a meta-algorithm used in ensemble learning. In every iteration of its main loop, it weighs more those attributes that contribute most to the error [3]. This technique allows for a reduction in the bias contribution to the error. KNN is a lazy algorithm since it puts the training dataset in memory and when it needs to make a new prediction, it searches all the training dataset to find the most similar instance in its neighborhood. Euclidean similarity function was used to measure the distance among tweets. The class to which every new tweet belongs to is decided by votation [3]. RF is a meta-learning approach that uses multiple random decision trees as base learner. The main characteristic of RF is that it contributes in reducing the influence of the variance in the total error. This is due to its selection method among different decision trees, each one of which is trained on a random sample with replacement based on the original data,

and the fact that it optimizes on a subset of the set of attributes in every step, rather than the entire set [3]. CSC incorporates arbitrary misclassification costs into the learning process. Misclassification penalties are associated with each of the four outcomes of a (binary) confusion matrix, referred to as CTP, CFP, CFN and CTN. TP (True Positive) is the total number of correct positive classifications, TN (True Negative) is the total number of correct rejections, FP (False Positive) represents the total number of misclassified instances that were incorrectly classified as positive, and FN (False Negative) is the proportion of positive instances that are wrongly diagnosed as negative. No costs are usually assigned to correct classifications, so CTP and CTN are set to 0. Since the positive class is often (and in our case) more interesting than the negative class, so CFN is generally greater than CFP [4]. Thus, the main objective of CSC is to minimize the expected overall cost as a function of the two error types, given by: Cost=CFP*FP+CFN*FN. Nonetheless, an unsupervised learning approach was also used in one of our experiments to compute the topics found in our tweet dataset. Non-negative Matrix Factorization is deemed as a clustering algorithm, where each cluster represents a topic. By using a document-term matrix containing some measure of the importance each word has in each document, it finds two new non-negative matrices as its factors. Those matrices contain a new dimension of unknown factors, also called latent variables. In this project, the documents are tweets and the latent variables are topics [5], [2]. We circumscribe the problem of spam detection on news media Twitter accounts. Thus, being able to detect spam in this context provides support in uncovering false news. In particular, we want to determine whether the techniques used by previous studies [6], [7], and [8] could be applied and proved in our case. The rest of the paper is organized as follows. The methodology and the proposed approach architecture are described in section 2. Section 3 presents and discusses the results. Section 4 presents related works in spam detection in social media. Finally conclusions and future works are given in the last section.

II. Methodology

The proposed approach is built to detect the presence of spams in Twitter stream. Fig. 1 illustrates a general block diagram of the workflow employed to construct the new classifier.

A. Dataset

3000 tweets [1] were downloaded from the three most popular Twitter news accounts in Venezuela: CaraotaDigital, la_patilla y CNNEE (CNN En Español), by using Birdwatcher [9]. From this dataset, we took a random sample of 946 tweets and labeled them by hand with two categories: Spam or Non-Spam. From this hand-labeled dataset, 700 tweets were used as training set [2] in some experiments, while the remaining as testing set. The whole dataset containing 3000 tweets was utilized later in the topics discovery step.

[1]https://github.com/J0hnG4lt/TSD/blob/master/Original.csv
[2]https://github.com/J0hnG4lt/TSD/blob/master/Training.csv

B. Pre-processing Phase

The preprocessing phase is considered as the most important phase in classifying text, since it has a very crucial impact on the computational performance of any ML algorithm. In this phase, three major tasks were applied: Feature extraction, normalization and labeling.

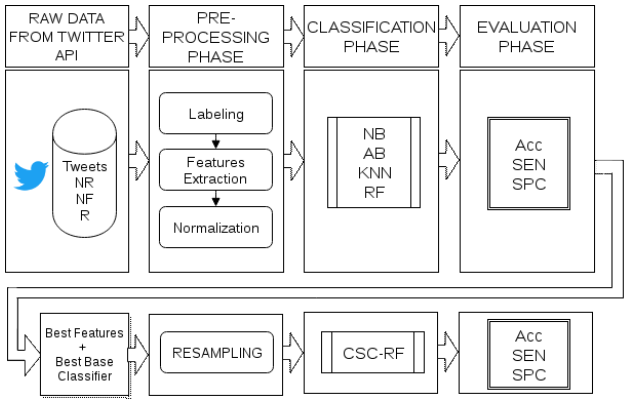

Fig. 1. Diagram of the workflow employed to construct the new spam classifier.

1) Labeling: Manual labeling technique was carried out by following the guidelines mentioned by Gordon [8]. For example, tweets with many uppercase letters or content related to pornography was deemed as spam. However, instead of classifying tweets into different types of spam, just as Gordon did in [8], we annotated binarily each tweet for spam or non-spam.

2) Feature Extraction: Ten features were extracted from the raw tweets and they are :

- Number of hashtags in text (NH) represents the total - number of times that the symbol # appears in a tweet.
- Number of mentions in text (NM) represents the total number of times other users have been mentioned in a tweet.
- Number of letters in uppercase (NLU) indicates how many letters in uppercase found in a tweet.
- Number of non-alphanumeric characters (NNC) in text, taking into account non-visible characters and whitespaces, reveals how many non-alphanumeric characters are present in a tweet.
- Number of URLs (NU) indicates the total number of external links (URL) found in a tweet.
- Length of the tweet (LT) refers to the number of characters in a tweet.
- Number of rational numbers (NRN) that are not part of any string, like a URL.
- Number of retweets (NR) refers to the number of times that a certain tweet's content gets reposted by different users.
- Number of favorites (NF) refers to the number of people that select a tweet as a favorite one.

- Topic (T) indicates the topic assigned to each tweet. The raw text of each tweet was preprocessed applying the following steps:
    - Tokenize each tweet into words, hashtags, punctuation marks, and special symbols.
    - Remove Stopwords. Stopwords are words which do not contain important significance to be used.
    - Map characters into their nearest ASCII character.
    - Convert the text to lowercase.
    - Remove accent and maintain the original vowel.
    - Apply non-negative matrix factorization algorithm (NNMF).
- Retweet (R) indicates whether a tweet has been retweeted or not.
- Vector Space Model (VSM): Term Frequency inverse document frequency (TF-IDF) was used to convert the textual representation of information into VSM (words vector). It is a measure of the importance of each word in a document and it reduces the effect of words that are very common [2]. It is calculated as in (3), where xi j is the number of times a word j appears in a tweet i, and N is the number of tweets. I is an indicator function that is equal to one when its argument is true.

$$tf(x_{ij}) = \log(1 + x_{ij}) \tag{1}$$

$$idf(j) = \log\Big(\frac{N}{(1 + \sum_{i=1}^{N} I(x_{ij} > 0))}\Big) \tag{2}$$

$$tfidf = \Big[tf(x_{ij}) * idf(j)\Big]_{j=1...V} \tag{3}$$

3) Normalization: Feature vectors were normalized scaling their values to a small specified range that varies from 0 to 1. Each value of a feature vector was divided by the sum of all the values of their features. The normalization step was applied to prevent samples with initially large ranges from outweighing.

4) Handling Imbalanced Class Problem: One of the techniques used to deal with imbalanced datasets is oversampling of the class underrepresented in them. This can be done by Resampling the data with Replacement (RWR), that is to say, every time an instance is chosen from the original dataset, it is returned to it and thus may be selected again [10].

C. Classification Phase

Five algorithms were used in the classification phase. They were trained and tested separately on training dataset, using 10-folds stratified cross validations. These algorithms were NB, Adaboost, KNN, RF and CSC.

D. Evaluation Phase

ML algorithms' performances were evaluated by using a set of metrics, which are accuracy (Acc), sensitivity (SEN) and specificity (SPC) [7]. We payed close attention to SEN, since our target attribute, Spam, is asymmetric. This means that, although the accuracy of many model configurations may be fairly good, they could still be useless if their SEN is not high enough. This problem is aggravated by the class imbalance present in the data, given the fact that genuine instances of tweets considered spam are less in number than the healthy ones.

Accuracy refers to the percentage of correctly classified tweets and it is given by (4) [3], where

$$Acc = \frac{TP + TN}{TP + TN + FN + FP} \tag{4}$$

SEN is equivalent to the true positive rate, and it represents in (5) the proportion of positives which are correctly classified [7].

$$SEN = \frac{TP}{TP + FN} \tag{5}$$

SPC is also known as the correct rejection rate, and it is defined in (6) as the proportion of negatives which are correctly classified [3].

$$SPC = \frac{TN}{TN + FP} \tag{6}$$

III. Results and discussions

Several experiments were conducted to evaluate the effectiveness of the new proposed twitter spam classifier. They were performed using four new sub-datasets that were built from the training set shown in Table 1:

- The first sub-dataset (D1) contains the first group of attributes, which are : NH, NM, NLU, NNC, NU, LT and NRN.
- The second sub-dataset (D2) holds the same attributes of D1 plus two new attributes, which are: NR and NF.
- The third sub-dataset (D3) comprises D2 plus an additional attribute that is related to tweet's topic (T).
- The fourth sub-dataset (D4) holds D2 plus two new features: R and VSM.

The following subsections present and discuss the results related to the process of dataset labeling, topic discovery and the construction of the proposed spam classifier.

A. Dataset labeling

We developed a script using Python language that allowed us to display each tweet's text on the terminal and then assign it a label. This, actually, was the same procedure employed by the authors in [8]. As an example, we show the following tweet:

> EL CUERPOTE: Serena Williams lo muestra en la Sports Illustrated 2017 https://t.co/y6AAeCBZ3w https://t.co/wH7r2lJsLb.

The features extracted from this tweet were: Zero for NH and NM, 1 for NRN, 22 for NLU, 11 for NNC, 2 for NU and 117 for LT. This tweet was considered spam because it presents a high number of NLU and its content fell within one of the categories of spam given by [8].

The distribution of tweets over class for both training and testing sets are illustrated in Tables 1 and 2 respectively. As can be noticed that 136 tweets out of 700 in training set were tagged manually as spam, while the remaining as non-spam. For testing set, 45 tweets out of 246 were labeled as spam.

TABLE I. CLASS DISTRIBUTION OF THE TRAINING SET

spam	ham	Total
136	564	700

TABLE II. CLASS DISTRIBUTION OF THE TESTING SET

spam	ham	Total
45	201	246

Two word clouds were built using the combined training and testing datasets. Fig. 2 and Fig. 3 show the most important terms related to spam and non-spam tweets respectively. They reflect that there is a qualitative difference between the two classes: spam tweets tend to be related to social events, content directed at adults, and movies, while non-spam tweets reveal an interest in the domestic politics from different countries. However, there are words appearing in both Fig. 1 and Fig. 2. This reflects the overall attention some controversial news received without necessarily comprising a topic directly related to spam.

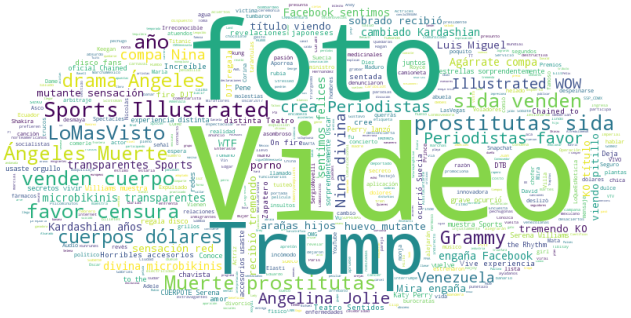

Fig. 2. Word cloud related to spam tweets.

B. Topic Discovering

To reveal the implicit knowledge present in news streams and to improve the classifiers' performances, we tagged each twitter by its main topic. The topics found in Table 4, reveal the particular reality from the time in which the tweets were being created. Most of them reflect the political situation found in the American Continent or simply what was popular at that moment. Additionally, we can noticed that the highest number of spams (34) are related to the topic which is linked to the current Venezuelan crisis, while the lowest (2) is related to Homicide case.

Fig. 3. Word cloud related to non-spam tweets.

TABLE III. CLASS DISTRIBUTION OF THE TESTING SET

topic	Bag of Words	#	Spam	Non-Spam
Interview given by a journalist	video, chavista, periodista, sentada	75	34	41
CNN TV Show about Venezuela	Venezuela, vivo, CNN, #soyfdelrincon, señal	115	18	97
US President's statements about México	Trump, Donald, mes, opinión, México	102	18	84
Prostitution	muerte, dólares, ángeles, prostitutas, venden	17	6	11
Ecuadorian presidential elections	Ecuador, vuelta, CNE, #lomásvisto, resultados	106	5	101
Immigration into the US	EEUU, indocumentado, inmigrantes, Aissami	72	6	66
Sports	Sports, Illustrated, 2017, rubia, portada	28	11	17
Old fashion trends	fotos, #lomásvisto, años, accesorios, orgullo	108	32	76
Homicide case	muere, brutal, embarazada, compañeras, recibir	41	2	39
Political Assassination in North Korea	Jong, Kim, Nam, muerte, Corea	36	4	32

C. Building Basic Spam Classifier

In this subsection, we conducted four different experiments employing four different ML algorithms for spam detection and classification. The goal of these experiments is two folds: i) discover the best combination of features and ii) determine the best base algorithm for the meta-classifier CSC. The following points can be observed from Table 5:

- In experiments I, II and IV, KNN classifiers yielded the highest rates of sensitive (44.10% and 49.30%) for spam class in comparison with AB and RF classifiers.
- AB classifiers produced the worst results to detect tweets as spams, but they were good for non-spam ones.

TABLE IV. RESULTS OF DIFFERENT EXPERIMENTS

Experiment		I	II	III	IV
Dataset		D1	D2	D3	D4
NB	Acc (%)	72.42	41.42	48.85	50.57
	SEN (%)	40.40	84.60	81.60	83.60
	SPC (%)	80.10	31.00	41.10	42.80
AB	Acc (%)	78.85	81.85	82.57	19.4
	SEN (%)	11.80	16.20	32.20	19.4
	SPC (%)	95.00	97.70	90.00	96.5
KNN	Acc (%)	77.28	77.00	76.71	83.00
	SEN (%)	44.10	44.10	42.80	49.3
	SPC (%)	85.30	84.90	85.40	91.00
RF	Acc (%)	81.28	74.00	85.00	86.42
	SEN (%)	26.50	36.0	41.20	50.3
	SPC (%)	94.50	83.2	95.60	92.1

- Although NB classifiers exhibited the highest rates of sensitive in all experiments, their performances' accuracies were too low; therefore NB algorithm is not appropriate to be used in spam classification problem.
- Topic feature could help AB and RF classifiers to get higher rates of sensitivity. For instance, SEN rate obtained in experiment II for RF classifier was increased from 0,360% to 41.20% when topic attribute was added. On the other hand, Topic feature could not provide any importance to KNN classifiers because SEN rate was decreased from 44.10% to 42.80%.
- In all experiments, it can be noticed that there was a large discrepancy between high specificity and lower sensitivity, yielding high false negative rates.
- The best results were 86.42% of Acc, 50.30% of SEN and 92.10% of SPC, exhibited by RF classifier in Experiment IV; this means that VSM feature could improve RF classifier's performance. In addition to that, R feature helped classifier to distinguish between spam and non-spam tweets.

From the above observations, we can conclude that the poor scores of SEN may be due to the class imbalance problem. Therefore, the following experiment was proposed to solve it.

D. Dealing with Class Imbalance Problem

Two steps were carried out to mitigate the problem of overfitting by balancing classes and by applying cost-sensitive learning rule. In the first step, resampling with replacement algorithm was applied using D4. As can be noticed from table 5, that although RF yielded very good results during training phase that reach 94.42% of Acc, 94.90% of SEN and 94.00% of SPC, but it performed worse during classifying unseen tweets, obtaining 51.10% of SEN for spam class.

TABLE V. RF'S RESULTS AFTER APPLYING RESAMPLING ALGORITHM.

RF	Acc (%)	SEN (%)	SPC (%)
Training	94.42	94.90	94.00
Testing	85.77	51.10	94.00

In the second step, we used cost-sensitive learning rule by assigning higher cost to false negatives than false positives. Thus, the RF's outputs were adjusted by changing their probability thresholds (penalties-pt) from the default value of $pt = 0.5$ to $pt = 2/(2+1) = 0.666$. Table 6 displays the training and test scores of this step. As can be observed from this table, that CSC-RF classifier exhibits the highest SEN, SPC and Acc, in training, of all experiments we have had carried out. Additionally, the discrepancy between low SEN and higher SPC was mitigated yielding 72.30% and 77.90% respectively in classifying unseen tweets as spam.

TABLE VI. CSC-RF'S RESULTS.

CSC-RF	Acc (%)	SEN (%)	SPC (%)
Training	89.14	96.90	81.40
Testing	76.82	72.30	77.90

IV. COMPARISON WITH RELATED WORKS AND ANALYSIS

A number of related research projects, such as [6], [7], and [8], were found. Authors in [8] aimed at training a random forest model to use it later as part of an information system that would allow users to detect spam. As such, that model was constrained to be fast and easily integrable to any web browser. Their project was based on [7], where it was determined that random forests were better than other classifiers that are traditionally used in spam detection. Using user-based and content-based features, their results achieved a SEN of 95.7%. The most significant difference between our approach and [7] was the exclusion of user-based features, given that news media accounts were assumed to be healthy and impersonal.

One of the differences of our project with [8] was that we did not intend to include several classes of spam. In this project, our class is binary.

Moreover, the study in [8] defined two methodologies for the model to be trained. The first one comprised only one phase and its intent was improving the model's performance on binary classification into spam or non-spam. The second one comprised two phases and its goal was improving the model's performance on multi-class classification into several categories of spam. Given that we only intended to use binary classification, we did not need to split our classifications into different phases. Gordon also showed, although in a very constrained manner, how to use a network approach in spam detection by executing a social network analysis on spammers. However, [8] did not find satisfactory results with the said approach.

Our approach in this paper differed from [8] since we did not take into account attributes related with groups of tweets or with user accounts. We deemed this as reasonable, given the constraint that the user accounts from which we took the raw data were well-known Twitter accounts. Thus, extracting information related to the network in which these accounts are embedded, is of no use, since we knew that they were not spam users in general. We were interested in finding instances of spam in Twitter accounts believed to be delivering trustworthy information. Thus, the results of the Random Forest binary classifier trained by Gordon in [8], that only used tweet features, achieved a SEN of 78% with an Acc of 81.2%. That author dealt with the class imbalance problem by undersampling.

Finally, one of the reasons we intended to constrain the problem of spam detection within the context of news media contents is the need for stopping misinformation from spreading on social networks. The most similar research project with this goal was [6]. In that case, what was wanted was detecting spam in streams of tweets supposedly related to

natural disasters. It was found that spammers made use of Twitter trends related to natural disasters, that usually contain useful information for the people affected by them, to attain more retweets. They achieved this by including fake information about such natural disasters. In our case, we were not constrained to working with news content only related to natural disasters. This means that the process of manual annotation in our project was actually more difficult than [6]. While the criteria used in [6] were concrete, it was extremely difficult, and even subjective, to separate spam news from truthful information in our case. One significant difference was their use of features related to users, including location, followers, and contacts. The author also used three classes: Legit, Fake and Spam. Using the Random Forest classifier and the 2013 Moore Tornado dataset, the SEN was 90% and its Acc was 91.71%. This model discriminated legitimate tweets from non-legitimate ones. The non-legitimate tweets were either spam or fake ones.

V. Conclusion

It was conjectured that the types of attributes that could be included would have the most influence over any ML classifier, and thus datasets D1, D2, D3, and D4 were used in different experiments. Nonetheless, it was found that the high class imbalance held more influence over the results. After dealing with the class imbalance problem, we found that the CSC-RF was the best model that could be found with a SEN of 72.30% and an Acc of 76.82% for testing. Although we had found that NB had a SEN of 83.6%, but its Acc was fairly low with 40.40%. These results have to be evaluated within the context of a highly specialized problem, since it was not a general spam detection problem. Given the fact that news media accounts cannot avail themselves of the same strategies used by other spam users, lest they be shunned by their audience, constraints in the strength some features have were bound to appear.

Our main contribution to the field of spam detection was the application of traditional techniques used in this field within the context of news media accounts from Twitter. Usually, spam detection is done in a general way within the medium in which it is embedded. However, given the high specialization of spam described in [1], the approach followed in this project was justified. In addition to this, the class imbalance problem was handled with resampling and Cost Sensitive Classification, an approach that is usually applied in spam detection for email [11].

As a future direction for this project, a specialization of spam detection that could take into account the writing style of each news media account may be used. Thus, instances of spam could be seen as deviations from normal behaviour from otherwise healthy accounts. The reason for proposing this is that each one of these accounts has an userbase that agrees with their own style of writing. Thus spam would not be identified with a particular way of reporting news, but only with those traits that make tweets irrelevant, repetitious, and vexing for their users. In addition to this, sentiment analysis could be carried out to determine whether a given tweet is either objective or subjective, which could be used as a new feature. Allowing a large number of users to label tweets as spam or non-spam from their favorite news media accounts could also be pursued.

Acknowledgment

The authors would like to thank the students: Edward Fernández, Joel Rivas, and Leslie Rodrigues for their support.

References

[1] F. Brunton, *Spam*, 1st ed. Cambridge, Mass.: MIT Press Ltd, 2013.

[2] K. Murphy, *Machine learning*, 1st ed. Cambridge, Massachusetts: The MIT Press, 2012.

[3] S. Marsland, *Machine Learning*, 2nd ed. Boca Raton, FL: CRC Press, 2015.

[4] M. Nabhan Homsi, N. Medina, M. Hernández, N. Quintero, G. Perpiñan, A. Quintana and P. Warrick, "Automatic heart sound recording classification using a nested set of ensemble algorithms", in Computing in Cardiology Conference, Vancouver, BC, Canada, 2016.

[5] A. Riddell, "Topic modeling in Python — Text Analysis with Topic Models for the Humanities and Social Sciences", De.dariah.eu, 2015. [Online]. Available: https://de.dariah.eu/tatom/topic_model_python.html. [Accessed: 13- Mar- 2017].

[6] M. Rajdev and K. Le, "Fake and Spam Messages: Detecting Misinformation During Natural Disasters on Social Media", 2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), 2015.

[7] M. McCord and M. Chuah, "Spam Detection on Twitter Using Traditional Classifiers", Lecture Notes in Computer Science, pp. 175-186, 2011.

[8] G. Edwards, "Spam Detection for Twitter", Undergraduate, University of Edinburgh, 2015.

[9] M. Henriksen, "Birdwatcher", GitHub, 2017. [Online]. Available: https://github.com/michenriksen/birdwatcher. [Accessed: 13- Mar- 2017].

[10] P. Tan, M. Steinbach and V. Kumar, *Introduction to data mining*, 1st ed. Boston: Pearson Addison-Wesley, 2006.

[11] J. Gómez Hidalgo, M. López and E. Sanz, "Combining text and heuristics for cost-sensitive spam filtering", Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning -, 2000.

Georvic Tur is a computer science student at Simon Bolivar University.

Masun Nabhan Homsi is an Assistant Adjunct Professor of computer science at Simon Bolivar University.