User Classification on Online Social Networks by Post Frequency

Gabriel M. Tavares
State University of Londrina
Computer Science Department
CEP 86057-970
Londrina, Brazil
gabrielmrqstvrs@gmail.com

Saulo Martiello Mastelini
State University of Londrina
Computer Science Department
CEP 86057-970
Londrina, Brazil
mastelini@uel.br

Sylvio Barbon Jr.
State University of Londrina
Computer Science Department
CEP 86057-970
Londrina, Brazil
barbon@uel.br

ABSTRACT

This paper proposes a technique for classifying user accounts on social networks to detect fraud in Online Social Networks (OSN). The main purpose of our classification is to recognize the patterns of users from Human, Bots or Cyborgs. Classic and consolidated approaches of Text Mining employ textual features from Natural Language Processing (NLP) for classification, but some drawbacks as computational cost, the huge amount of data could rise in real-life scenarios. This work uses an approach based on statistical frequency parameters of the user posting to distinguish the types of users without textual content. We perform the experiment over a Twitter dataset and as learn-based algorithms in classification task we compared Random Forest (RF), Support Vector Machine (SVM), k-nearest Neighbors (k-NN), Gradient Boosting Machine (GBM) and Extreme Gradient Boosting (XGBoost). Using the standard parameters of each algorithm, we achieved accuracy results of 88% and 84% by RF and XGBoost, respectively.

CCS Concepts

•**Computing methodologies** → **Supervised learning by classification;** •**Information systems** → *Data mining;* •**Applied computing** → Document management and text processing;

Keywords

Online Social Networks; Machine Learning; User Classification; Twitter

1. INTRODUCTION

Online Social Networks (OSN) are online environments where there is an interaction between people in general with different objectives [12, 13, 22]. These interactions are characterized by the creation of bonds and connections based on exchange of opinions and interests in common. Since OSN are so broad and cover a huge number of users, they attract enterprises attention either to announce their product or to analyze how their brand is received by the public opinion [3, 17, 21].

At the same time, OSN have become an interesting environment for users with malicious purposes. The immense number of users and the difficulty faced by OSN to combat mischievous behavior bring even more space for wrong actions. Twitter in special is an interesting target for bots that want to spam, phish, or highlight the popularity of a determined subject by the use of hashtags. The OSN chosen for this work was Twitter because a high number of bot accounts can be found in it and spam is a systemic problem [18]. Twitter has an easy concept and a user friendly interface, its posts have the maximum of 140 characters (popularly known as tweet) and it allows images and links. Differently from others OSN, Twitter has no conversation based on individual chats.

OSN in general face problems to ban malicious accounts. Usually there is no automatic ban, that is, no method or algorithm is used for account banning on a large scale. This happens mostly by the fact that automation may ban some legitimate accounts and, by doing that, the OSNs popularity will eventually decay in the public opinion. Some OSN have been implementing features such as account reporting. This tool is focused on the user side since its principle is that users will report malicious accounts when they come across one. However, this approach does not reflect positively since most users are not in the OSN for the purpose of reporting malicious accounts and it can be said that great part of users do not care to report when they come across such accounts [11]. Adding to that fact, there is the misuse of the reporting tool, it is common to detect groups of people that report other accounts driven by the difference of opinions, this is mostly seen in political matters.

There have been attempts to overcome this matter in some works. Chu [7] has an approach with several pre processing techniques. His work uses not only text in the analysis, but also frequency values and account related information. His goal is also to classify users as Humans, Bots or Cyborgs. The core of the work is subdivided in four steps, they are: an entropy based in posts interval, for that the author uses all tweets from the account; a machine learning algorithm using the content of tweets to detect spam with Bayesian classification; an account properties component, with infor-

SBSI 2017 June 5^{th} – 8^{th}, 2017, Lavras, Minas Gerais, Brazil

mation such as the URL ratio in the tweets, the device that was used for tweeting, followers to friend ratio and more; the last step is the decision maker with a Linear Discriminant Analysis that uses the 3 initial steps as input for the decision and posterior classification. Chu's work concluded that entropy, URL ratio and the device used for tweeting were the most important descriptors for an account. However, since the work uses so much information, it gets very tied up to Twitter's structure and it would be hard to apply that approach to other OSN and even on Twitter itself, given that the micro-blog is constantly changing. Also, the approach uses deep pre processing techniques that makes the problem even more complex and hard to replicate.

A different approach was proposed by Igawa [14], where a wavelet-based approach was used for account classification that detects bot dissemination in OSN. This work uses pure text mining solution, that is, no other descriptors were used. A new weighting scheme was proposed with different configurations. At last, a concern about computational complexity was raised. As the previous explored work, this one also is filled with several pre processing steps and techniques, which raises the complexity and the difficulty to replicate.

Our approach is focused only on frequency analysis. The main advantage is that it can be applied in any OSN, since all of them notes the time when the post was done. In other words, our method could be applied in OSN of picture, video, music or other media beyond textual-based OSN. Adding to that, the process is free of pre processing techniques. The descriptors are extracted directly from the post's time and the classification process can be performed after that. The goal is to identify malicious accounts and help OSN find and ban those accounts. The classes are Human, Bot or Cyborg.

2. MATERIALS AND METHODS

The following section discusses the acquired basis, as well as the methods that were used in the development of this work. The methodology can be separated in five general steps: Acquisition, Feature Extraction, Feature Selection, Classification and Experimental Setup.

2.1 Materials

The environment chosen for development was RStudio, based on the R language version 3.3.3. This choice was due to its wide use in statistics and data mining. It also offers a broad range of packages for Machine Learning. To parametrize all the algorithms the package CARET (classification and regression training) was used. This package offers support to various algorithms and its parameters can be set easily. It contains functions to streamline the modeling process for complex regression and classification problems, using several R packages.

One of the most basic tools of the CARET package offers is the *train* function, which can be used to evaluate, by means of resampling, the effect of the adjustment parameters of the model in the performance, choose the optimum model given the parameters and estimate the model performance from a training base. It is important to note that there are customization options for virtually all parameters for each classification algorithm, this increases the performance of the chosen algorithms since the best possible cases can be select.

2.2 Dataset Setup

This section discusses the acquired data set. As the analyzed data came from Twitter, an extraction of tweets was required. The extraction of the samples was done through the Twitter API, which provides access to reading and writing data on the micro-blog. Thus, 99 human accounts, 42 bot accounts and 90 cyborgs accounts were mined. The classification of these accounts was done by a specialist based on proposed method on [7], making possible, therefore, the development of the next steps of this work. For each account, approximately 200 tweets were acquired.

The tweets came in with little information, they were: *user*, *timestamp* and *tweet*. The *user* field contains the account name set for that user that was mined, this is simply used for identification purposes and has no effect on the implemented methodology. The *timestamp* column represents the time and date that the tweet was submitted by the user in the micro-blog. An example of this data is "10/09/2015 06:54:14", which means that this respective post was acknowledged at six o'clock in the tenth of September of 2015. All kinds of features can be extracted from the *timestamp* values and this will be more deeply explored in the section 2.3. At last, the *tweet* field was the actual text that was posted on Twitter. An example of a *tweet* is "I remember when the Seahawks were not very good and the fans following online were ecstatic with wins and took losses in stride.". In this case the content is clear and unambiguous, it has intelligence in it and most probably it belongs to the human class. Most works that attempt to classify users in OSN use this field as the parameter for analysis. However, this work brings a new approach to this matter that uses only frequency attributes, thus, the *tweet* content for all users was ignored. There was no need to extract extended information about the user account and its posts, that is, country, language, time zone, profile image, friends, followers, among others were all ignored from the API extraction.

Later the respective class was added manually for each account based on its behavior. According to [7] humans are characterized by non automatic actions, with original, intelligent, specific and human-like contents. Also, a human user usually talks about what he is doing at the moment or what he feels about something. Actually, this behavior is seen as human because humans use Twitter as a tool to display themselves and interact with friends. Bot accounts, which is a reference to the English language word robots, are characterized by automatic behavior and their purpose is varied including posting of spam, the practice of pishing through malicious links and the attempt to increase the popularity of a determined subject, brand or product. In Twitter, bots have the goal of behaving like humans to gain followers and create a network where their activities can be disseminated [20]. Thus, bots are characterized by the lack of originality, excessive automation and high presence of spam and URL links in their texts. Lastly, cyborgs are the intermediate between the previous classes, it has both automatic and non automatic behavior. Sometimes it can behave intelligently like a human, but at other times it presents automatic updates of RSS feeds. This means that there is a human tweeting and creating content, but when the human is not there, the account keeps its updates, either by the use of RSS, API, apps or other types of automation. An example of cyborg account is a journalistic profile, where some posts are done directly from Twitter and other posts are done by the use

of the API, which is classified as an automatic post.

Following the presented definitions, a careful manual classification was performed. Each account was explored in the micro-blog, its posts verified and its behavior analyzed and then a class was attributed to an account. This way the next steps of the method could follow.

2.3 Features Addressed

After the account classification, the next step was to extract the frequency values from the accounts. The aiming of this step is to describe the classes with attributes the best way possible, so the classifier can be able to learn the classes behavior and predict new samples. The total of 34 descriptors were extracted, they were:

- Average interval between posts in seconds (AIP);
- Average posts per day (APD);
- Average posts per week (APW);
- Histogram of posts per hour (HPH);
- Histogram of posts per day of the week (HPD).

This descriptors can map in completeness a user behavior and its posts frequency. The AIP, APD and APW represent a single value, while HPH, as a histogram, represents 24 values (from 24 hours) and HPD 7 values (from 7 days).

Since there are several features, one feature selection technique was implemented to filter the most relevant ones. The chosen technique was the χ^2 test of independence. The χ^2 distribution is one of the most used distributions in inferential statistics. This test serves to quantitatively evaluate the relationship between the outcome of an experiment and the expected distribution for the phenomenon. That is, it tells us with certainty that the observed values can be accepted as governed by the theory in question [16].

The χ^2 outputs each variable from 0 to 1 in importance. The most relevant variables are explored in the section 3 later on this work. For now, the goal was to identify the non important descriptors and eliminate them from the classification process. Fortunately, there was only one descriptor that obtained 0 from the χ^2 test, this feature was APD. All other features had relevant results and could not be ignored in the classification process. Since only one feature was not well classified in the χ^2 test, we chose to maintain this descriptor in the next steps of this work. This decision was due to the small number (only one) of non important descriptors identified, that is, the complexity of the training and classification process would not be affected by only one more descriptor.

Another observation that can be drawn from the statistical analysis is that almost all features were relevant to describe the account's characteristics. That is, this work was able to extract the most important and relevant features and that will aid the classifiers and enhance the general performance.

Regarding APD as being a non relevant feature, that is probably because most accounts can be classified either by the average interval or, mostly, by the time they use to make their posts. APD little matters because the three classes have examples of users with lots of posts per day and at the same time users with few posts per day. Thus, APD by itself is not able to correlate with the account's class.

2.4 Classification Approach

According to Dougherty, Machine Learning (ML) is the field that looks for algorithms that allow the computer to recognize patterns such as the distinction of numbers or faces. Given the certain descriptors, it is possible for an algorithm to learn and be able to classify samples related to the problem in question [9]. ML is concerned on how computers can auto-program and infer information from data [5].

The final goal of this paper is to classify users in OSN based in frequency of posts. Thus, it is interesting to apply various machine learning algorithms that come from different paradigms to explore contrasting possibilities and evaluate the better ones. The follow list details the chosen algorithms:

- Random Forest (RF): RF are an ensemble learning method for classification and regression composed by decision trees proposed by Breiman in [4]. It overcomes the overfitting problem by training the trees with random attributes. The class comes as the result of the most voted class by the majority of trees. RF are considered robust to errors and outliers are efficient in big data sets.
- Support Vector Machine (SVM): SVM uses supervised learning, where the training learn the class characteristics based on the class label. The model built represents the examples as points in space and different classes are divided by a gap that separates the points. SVM has a well defined statistical base and it has a great capacity of generalization [19].
- K-Nearest Neighbors (k-NN): In pattern recognition k-NN can be used in classification and regression. Its functioning is to discover the k closest neighbors and for classification, it classifies the element based on the class that most appears on the k neighbors. k-NN is a type of instance-based learning and it is one of the most simple machine learning techniques [8].
- Gradient Boosting (GBM): GBM is a widely acclaimed technique for building predictive models. It produces its models in the form of an ensemble of weak prediction models, usually decision trees. Briefly, GBM involves a loss function, a weak learner and an additive model that adds weak learners to minimize the loss function so the classifiers are built one at a time and they try to correct the previous ones errors [10].
- Extreme Gradient Boosting (XGBoost): XGBoost is an implementation of gradient boosting decision trees designed for performance and also is an open-source software library which provides the gradient boosting framework for several programming languages. Its design is focused in high efficiency, flexibility and speed [6].

In most cases, ML falls into two categories of learning: supervised or unsupervised. In the first case the algorithm learns from a database that is already previously labeled, that is, the examples that the algorithm analyzes already have a defined class. In the unsupervised approach the data does not have labeling, so the algorithm has to analyze the behavior based only on its attributes [15]. There is also a

hybrid between the two called semi-supervised learning, in which case the training set has some of the data labeled. As presented in previous sections, all the data in this work was already labeled before the classification phase. That is, the ML algorithms learnt the classes behavior from actual examples from each class, which classifies this work's methodology as supervised learning.

In supervised ML, the classification can be applied to binary problems (two classes) or multi-class problems (more than two classes). Many real problems in areas such as medicine, bio informatics and computer vision translate into multi-class problems. A binary classifier is simpler than a several classes classifier, because in contrasting several classes, the attributes that describe them are more sensitive and this may cause a decrease in accuracy. However, there are situations where the classes can not be reduced to only two and then the multi-class approach is needed.

Thus, this work proposes a multi-class classification of OSN accounts using supervised learning and frequency as the accounts descriptor.

2.5 Experimental Method

This section presents how the tests were prepared and also the metrics used to evaluate the proposed methodology.

2.5.1 Evaluation Metrics

In ML a confusion matrix is a table that allows performance visualization and measurement of an algorithm [1]. In a binary analysis, it is called True Positive (TP) and True Negative (TN) the instances correctly classified as positive or negative, respectively. False Positive (FP) represents the number of instances that were wrongly classified as positive and False Negative (FN) the number of instances that were positive, but were classified as negative.

Table 1: Result of a classifier for a binary example problem

i	r(i)	p(i)
i_1	P	P
i_2	N	P
i_3	N	N
i_4	P	N
i_5	N	P
i_6	N	N
i_7	P	P
i_8	P	P
i_9	N	N
i_10	P	P

Table 2: Confusion Matrix from Table [1]

	P	N
P	4(TP)	1(FN)
N	2(FP)	3(TN)

Table 1 is a common example of the outcome of a classification process. The first column identifies the items, the second column shows the real class of the element and the third column is the prediction of the classification algorithm. From this a confusion matrix can be created (Table 2) which illustrates the values obtained from the prediction. The main diagonal of the matrix has elements that have been correctly predicted. For educational purposes, this example shows a binary classification, however, a multi-class confusion matrix can be constructed using the same principles shown here.

Several measures can be drawn from a confusion matrix and this work uses some of them. The first one is accuracy because it is a widely used performance metric in ML [2] which represents the proportion of instances predicted correctly in relation to the total of predicted instances. The calculation of the accuracy is given by the following equation:

$$Acuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Next measures are precision and recall. Precision show how correct and relevant the results are while recall is the fraction of relevant documents that were retrieved.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

At last, F_1 score is the weighted average of precision and recall and it varies from 0 to 1.

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{4}$$

Following the Tables 1 and 2, the metrics results are:

- Acuracy: 70%
- Precision: 66,67%
- Recall: 80%
- F_1: 0,72

2.5.2 Tests

All the presented algorithms were tested in the same environment so that the results can be compared. To perform tests some parameters needed to be defined beforehand. The first one is the holdout, where 70% of the base was used for training the classifier, while the remaining 30% was used for testing. Thus, the separation of the basis between test and training can be done, 70% of each data set representing a class was withdrawn and joined to the respective 70% of the other class, thus being possible to create a concise base of data that is subsequently scrambled. In the same way the database for testing was created, it is important to note that this base consists of the rest of the samples that were not used in the training, that is, a sample is either in the test or the training and never in both at the same time. All classifiers used 10 fold cross-validation.

In the next step, each of the classifiers performs their respective training. The training process generates a model for the classifier, this model is built with the knowledge obtained from the supervised learning. Once the model is constructed, new samples are tested. These new samples are from the testing data set, their class is hidden so the model will classify each sample based only in its frequency

attributes. Only then it was possible to create a confusion matrix for each of the methods and to measure the effective accuracy of each case.

In order to explore all possible combinations, the entire process described was executed 50 times for each classifier. A multi-class test was applied in all cases: Human vs. Bot vs. Cyborg. This test shows the strengths and weaknesses of each classifier, as well as pointing out which classes have the closest behaviors, which are therefore harder to differentiate.

It is important to mention that during the training step, all algorithms were submitted to the CARET package with the default parameters and no further exploration and enhancement of specific algorithms was performed. That is, there was no tuning for any algorithm, this decision was taken with the goal of maintaining a fare comparison between them.

3. RESULTS

The first performance measurement analysis of the different classifiers was done from the accuracy that each obtained in different cases. In general, as Figure 1 shows, the Random Forest classifier obtained a superior mean accuracy with 87.7% and was also the classifier with the least variance, although it had a few outliers. Both XGBoost and GBM followed closely with 84.3% and 83.9% respectively. At last, SVM and k-NN were the last accurate classifiers, with 78.9% and 72.8%.

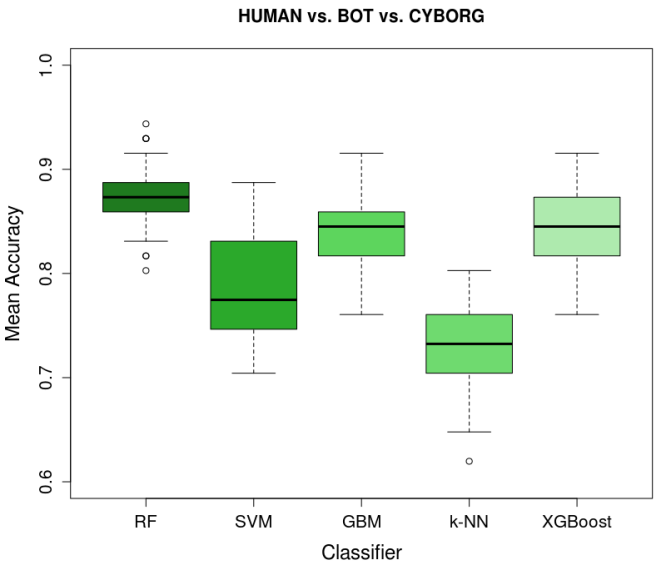

Figure 1: Classifiers Accuracy

RFs flattened quartiles show that the classifier can maintain its high performance by having a good sense of abstraction, independently from its training base. On the other hand, SVM even shows high accuracy values, however, it fails to maintain this behavior constantly, this means the classifier is highly dependent of the training set.

In [14], the author used several configurations for classification, and his best accuracy for the multi-class experiment among humans, cyborgs and bots was 91%, with an average of 87.5%. This analysis was based solely on textual features. Chu [7] reaches 90.5% of true positives with a hybrid approach, taking mostly into account textual features, with several pre processing steps and an analysis very tied up to Twitter. On the other hand, this work was based on frequency data analysis. This approach is simpler and more practical because it does not have to deal with problems inherent to text, as other works have faced. The use of frequency data, however, did not negatively affect the performance of the classifiers, on the contrary, accuracy measurements got a mean of 87.7% with RF. Also, adding to that, this approach can be translated into any OSN because the information used as account descriptor was only the frequency. Thus, this classification can be performed in audio, image, video and text OSN.

Figure 2 shows the comparison between the metrics and classifiers. This graph can explore more deeply all algorithms' behavior and how it affected classification. As seen in accuracy mean, RF obtained higher values of precision, recall and F_1 score. Some conclusions can be extracted from this. First, RF is the most precise classifier, i.e., from the elements labeled as the positive class, 86.5% of it indeed were. Second, A high recall value means that from all elements contained in the true positive bucket, 84.1% were correctly retrieved and classified. At last, the F_1 score, which is the weighted average of recall and precision, shows that the RF maintained a healthy relation between its metrics.

All metrics from Figure 2 take true positives into account and this is the most important value in account classification in OSN. This is because OSN face difficulties banning malicious accounts and a high value of false positives would implicate that a high number of benign accounts were being classified as malicious. As stated before, OSN try to come around this problem since banning legitimate accounts can decrease the OSN's popularity.

From the figures, it can be observed that the model induced by k-NN obtained the smallest accuracy compared to the others. The k value used for all tests was 5 since it shows a good balance between performance and computational cost, thus, in this case, the algorithm was a 5-NN. The need for specific parameters is the explanation for the low accuracy behavior. Since the beginning of this work's methodology, the use of the CARET package was defined by its comprehensiveness and ease. For the tests, therefore, all the algorithms were applied in their default configurations within the package. In this way, the initial settings for the k-NN made it behave not the best way possible. In order to maintain the strict and neutral approach, there was no improvement of parameters for any of the analyzed algorithms. Thus, the k-NN ended the lowest accuracy levels among the classifiers tested.

Figure 3 shows the established relationship between classes and the metrics proposed for analysis. The bot class clearly had the lowest performance in all metrics. Even though bots are automated, their frequency values can be sometimes mistaken as human or cyborgs. Also, bots tend to hibernate after some active time, this hibernation might mask the bot characteristics and some samples end up wrongly classified. Cyborgs had the highest precision, that is, this class is more rarely mistaken as the other two. Finally, humans had the highest recall and F_1 score, making this class the easiest to classify because humans have non automated ways of behaving, contrary to cyborgs and bots that are programmed to do a specific task and present clearer patterns of behaviors,

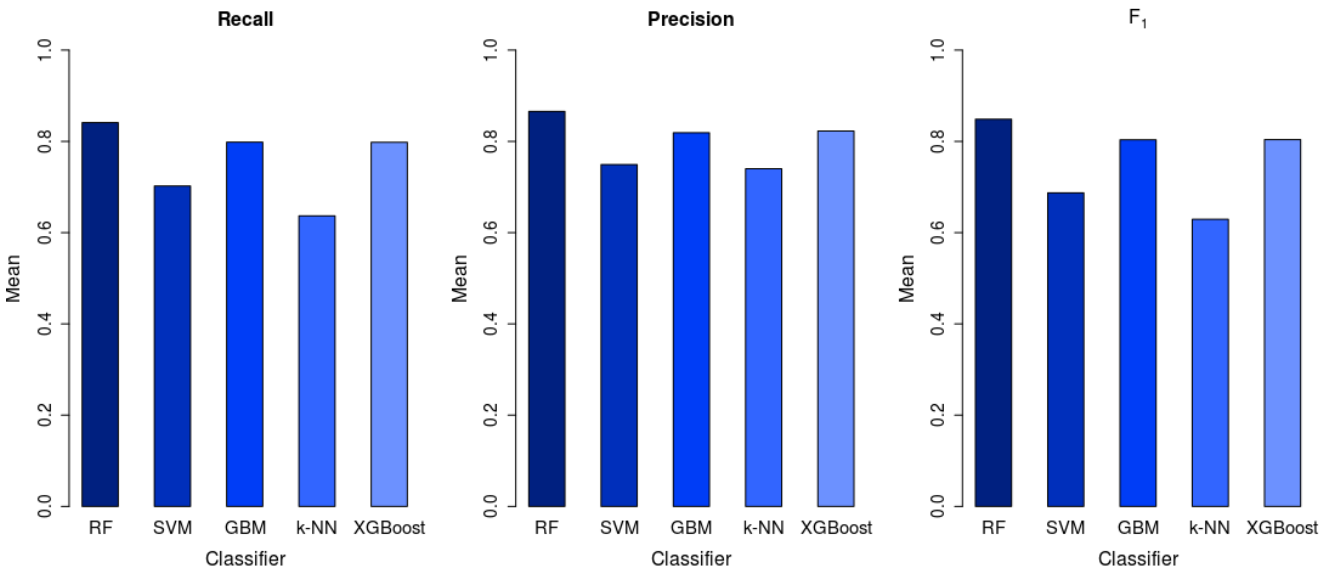

Figure 2: Metrics vs Algorithms

especially in relation to the frequency of posting

Table 3 shows the twentieth most important variables according to χ^2 test. For this, all dataset from all classes were combined, then the most influential features were selected. The χ^2 test can vary from 0 to 1, being 1 the maximum. It is obvious that the HPH descriptors are the most important for the ML training and classification, they occupy all positions from the first to the fourteenth. That is, the hour of a post is the feature that best describes the user behavior and the class can be drawn from that. Only after that, one HPD descriptor and AIP show up in the table. This means that the day of posting and the average interval between posts are less important for classification matters. Table 3, then, shows that the behavior that most indicates a user's class is the time they usually make their posts, much more than the day of the week or the average time between their posts

APD, APW and almost all HPD descriptors were not included in the twentieth most important attributes according to χ^2, meaning that those are not relevant enough to describe the classes. Both APD and APW are average values of posts, since all classes vary a lot in this characteristic, this descriptors ended up not interfering directly in the class of each account. HPD descriptors tell which day of the week the post was submitted to the OSN, they also are not very important because either automated accounts or non automated ones do not have a clear pattern relating posts and days of the week.

Lastly, the evaluation metrics showed in this section could present better results with tuning of the algorithms. However, this work is more concerned with the proposed technique and not necessarily with the performance of ML algorithms. Moreover, tuning is closely related with a specific dataset, which could possibly generate an undesired bias in the analysis of our technique.

Table 3: Importance of variables according to χ^2 test

Feature	χ^2 Importance
VAR_HOUR_22	0.62
VAR_HOUR_9	0.58
VAR_HOUR_21	0.57
VAR_HOUR_8	0.57
VAR_HOUR_10	0.56
VAR_HOUR_7	0.56
VAR_HOUR_6	0.54
VAR_HOUR_17	0.52
VAR_HOUR_19	0.50
VAR_HOUR_4	0.50
VAR_HOUR_2	0.49
VAR_HOUR_11	0.49
VAR_HOUR_20	0.49
VAR_HOUR_1	0.49
VAR_WEEK_DAY_TUE	0.47
AVG_INTERVAL_POSTS	0.47
VAR_HOUR_18	0.46
VAR_WEEK_DAY_MON	0.46
VAR_HOUR_12	0.46
VAR_HOUR_23	0.46

4. CONCLUSION

This work presented an alternative methodology for the classification of users in OSN using as basis of analysis the frequency of postings made by users, unlike traditional approaches, which are based on textual data to make their analysis. This type of approach illuminates a new area in fraud detection in OSN, which in the future can be combined with traditional methods to obtain greater accuracy.

The algorithms used for ML were RF, SVM, k-NN, GBM and XGBoost. The results of the tests following the proposed methodology showed in general that the RF has a greater accuracy in relation to the other classifiers. The problem was tested in multi-class classification (Human, Bot

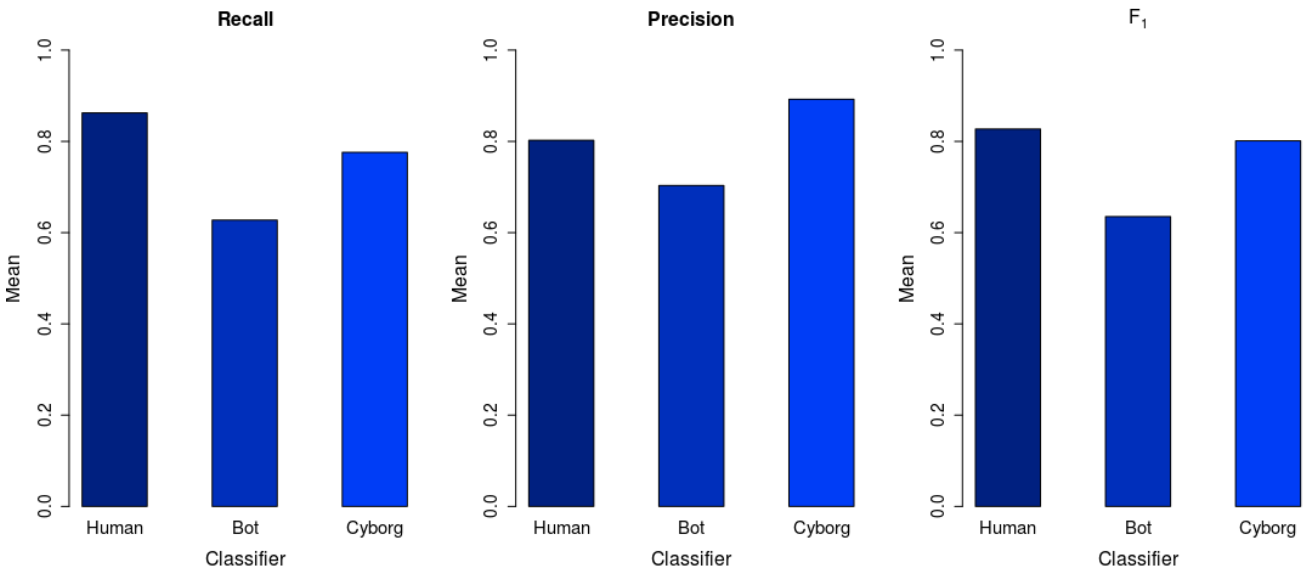

Figure 3: Metrics vs. Classes

or Cyborg). Thus, RF and XGBoost classifiers have maintained the best performances in multi-class experiments, this shows the excellence of these algorithms and their ability to adapt to the problem. The parameter adjustment was performed in an equivalent way for all the algorithms, thus maintaining a fair dispute between them, that is, no algorithm was changed or had improved parameters (all were submitted to the CARET package).

The most relevant descriptors for the classification were those related to the post time (HPH). Automated accounts have a predefined behavior that can be learned by the classifiers. It can then be deduced that the hours of activity in OSN is the most important factor in the identification of accounts. On the other hand, the day of the week (HPD) and the average of daily (APD) or weekly (APW) publications did not weigh heavily on the classification.

Thus, the frequency based fraud detection approach has proved to be effective and can address a gap found in the state of art. As future work, testing on other bases should be performed, in addition, new acquisition methods should be tested in order to obtain more frequency usage data. Also, as a next step, this work will be extended with the application of this technique on Data Stream Mining, where there is a stream of data that can be evaluated and decisions have to be taken with minimum delay time as possible.

5. REFERENCES

[1] C. C. Aggarwal. *Data classification: algorithms and applications*. CRC Press, 2014.

[2] E. Alpaydin. *Introduction to Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2004.

[3] S.-A. Bahrainian and A. Dengel. Sentiment analysis and summarization of twitter data. In *Computational Science and Engineering (CSE), 2013 IEEE 16th International Conference on*, pages 227–234. IEEE, 2013.

[4] L. Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, Oct. 2001.

[5] J. G. Carbonell, R. S. Michalski, and T. M. Mitchell. An overview of machine learning. In *Machine learning*, pages 3–23. Springer, 1983.

[6] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM.

[7] Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia. Who is tweeting on twitter: Human, bot, or cyborg? In *Proceedings of the 26th Annual Computer Security Applications Conference*, ACSAC '10, pages 21–30, New York, NY, USA, 2010. ACM.

[8] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, January 1967.

[9] G. Dougherty. *Pattern recognition and classification: an introduction*. Springer Science & Business Media, 2012.

[10] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 2000.

[11] S. Ghosh, G. Korlam, and N. Ganguly. Spammers' networks within online social networks: A case-study on twitter. In *Proceedings of the 20th International Conference Companion on World Wide Web*, WWW '11, pages 41–42, New York, NY, USA, 2011. ACM.

[12] A. Hassan, A. Abbasi, and D. Zeng. Twitter sentiment analysis: A bootstrap ensemble framework. In *Social Computing (SocialCom), 2013 International Conference on*, pages 357–364. IEEE, 2013.

[13] L.-C. Hsieh, C.-W. Lee, T.-H. Chiu, and W. Hsu. Live semantic sport highlight detection based on analyzing tweets of twitter. In *Multimedia and Expo (ICME), 2012 IEEE International Conference on*, pages

949–954. IEEE, 2012.

[14] R. A. Igawa, S. B. Jr, K. C. S. Paulo, G. S. Kido, R. C. Guido, M. L. P. Júnior, and I. N. da Silva. Account classification in online social networks with {LBCA} and wavelets. *Information Sciences*, 332:72 – 83, 2016.

[15] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated, 2014.

[16] K. Molugaram and G. S. Rao. Chapter 9 - chi-square distribution. In K. Molugaram and G. S. Rao, editors, *Statistical Techniques for Transportation Engineering*, pages 383 – 413. Butterworth-Heinemann, 2017.

[17] M. M. Mostafa. More than words: Social networks' text mining for consumer brand sentiments. *Expert Systems with Applications*, 40(10):4241–4251, 2013.

[18] K. Thomas, C. Grier, D. Song, and V. Paxson. Suspended accounts in retrospect: An analysis of twitter spam. In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, IMC '11, pages 243–258, New York, NY, USA, 2011. ACM.

[19] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[20] R. Wald, T. M. Khoshgoftaar, A. Napolitano, and C. Sumner. Predicting susceptibility to social bots on twitter. In *2013 IEEE 14th International Conference on Information Reuse Integration (IRI)*, pages 6–13, Aug 2013.

[21] S. J. Yu. The dynamic competitive recommendation algorithm in social network services. *Information Sciences*, 187:1–14, 2012.

[22] M. Zappavigna. Ambient affiliation: A linguistic perspective on twitter. *New Media & Society*, 13(5):788–806, 2011.