Improving Malicious Accounts Discrimination through a New Feature Engineering Approach Using Relaxed Functional Dependencies

Loredana Caruccio[1], Gaetano Cimino[1], Stefano Cirillo[1], Domenico Desiato[2,*], Giuseppe Polese[1] and Genoveffa Tortora[1]

[1]*Department of Computer Science, University of Salerno, via Giovanni Paolo II n.132, 84084 Fisciano (SA), Italy*
[2]*Department of Computer Science, University of Bari Aldo Moro, via Edoardo Orabona n.4, 70125 Bari (BA), Italy*

Abstract

Social network platforms include several tasks such as advertisements, political communications, and so on, producing vast amounts of data spread over the network. Consequently, verifying the truthfulness of such data and the accounts generating them becomes necessary. In particular, malicious users often create fake accounts and followers for harmful activities, potentially producing negative societal implications. In order to increase the capability to identify fake accounts correctly, this discussion paper presents a new feature engineering strategy that exploits relaxed functional dependencies (RFDs) to enhance the capability of existing machine learning strategies in discriminating fake accounts. In particular, experimental results conducted using several machine learning models on account datasets of both the Twitter and Instagram platforms emphasize the effectiveness of the proposed approach in fake account discrimination activities.

Keywords

Data management, Fake accounts, Data Profiling, Social networks

1. Introduction

Social networks enable sharing information among users of all ages, at every moment, and in every part of the world. Social interaction platforms like Instagram, Twitter, Tumblr, etc., have a significant impact on the daily life of their users and the entire society.

A fundamental aspect to be monitored over a social network is the popularity of a profile, witnessed by the number of its friends or followers. A Twitter or Instagram profile with many followers is considered influential, hence it provides a better reputation to the profile's owner and attract better-paid advertisements. Consequently, a common practice of several social network users is to buy fake followers to appear more influential, also because they can be bought at an extremely low price (a few dollars for hundreds of fake followers). If this practice was merely used to support individual vanity, it would be harmless, but if it aimed at making an account more reliable and influential, it might be dangerous.

SEBD 2024: 32nd Symposium on Advanced Database Systems, June 23-26, 2024, Villasimius, Sardinia, Italy
*Corresponding author.
lcaruccio@unisa.it (L. Caruccio); gcimino@unisa.it (G. Cimino); scirillo@unisa.it (S. Cirillo); domenico.desiato@uniba.it (D. Desiato); gpolese@unisa.it (G. Polese); tortora@unisa.it (G. Tortora)
0000-0002-2418-1606 (L. Caruccio); 0000-0001-8061-7104 (G. Cimino); 0000-0003-0201-2753 (S. Cirillo); 0000-0002-6327-459X (D. Desiato); 0000-0002-8496-2658 (G. Polese); 0000-0003-4765-8371 (G. Tortora)

In general, it is possible to find many types of anomalous accounts in social networks, such as Spammers, Bots, Cyborgs, and Trolls. Spammer accounts tend to recommend fake contents and/or dangerous links. Bot algorithms tend to manage accounts to simulate human behavior, trying to automatically perform typical human actions. With respect to them, Cyborgs are both managed by humans, hence they are not necessarily malicious. For instance, a politician might not handle his/her account personally and might rely on a stuff of people together with some Bots. Finally, Trolls are algorithms aiming to disrupt conversations and activities of others.

Related work. In the literature, we find several automatic techniques for identifying spams and bots [1, 2]. Some of them focus on the characterization of human behaviors with the help of sociologists, whereas others exploit supervised machine learning techniques on datasets containing different types of accounts, manually classified by humans [3, 4]. Additional work relies on the features of user profiles, and on those related to the behavior and timing of accounts, in order to identify spammers in microblogging, by employing multi-feature strategies [5, 2]. In our proposal, we focus our attention on the detection of fake accounts by trying to infer peculiar characteristics, in terms of correlations within the data of a social network profile dataset, aiming at enhancing the capabilities of machine learning methods to discriminate them.

In this discussion paper, we describe the feature engineering strategy proposed in [6] that exploits relaxed functional dependencies (RFDs) holding on fake accounts data to define new features for the data with the aim of improving performances of predictive models in discriminating fake accounts. Moreover, since the addition of new features could introduce noise and expose the machine learning model to problems, such as overfitting and underfitting [7], we also propose a novel FAV-based feature Evaluation Metric (FEM) for ranking the new features and select the most relevant ones.

The remainder of the paper is organized as follows. Section 2 presents the new proposed feature engineering strategy and the associated FEM metric. Section 3 presents experimental results, whereas Section 4 concludes the paper and provides directions for future work.

2. A Feature Engineering Strategy for Discriminating Fake Accounts

In general, although the addition of new features can potentially increase the training time of classification algorithms, it could lead to the creation of more concise and accurate classifiers. Moreover, meaningful features could contribute to the understanding of the learned concept [8], but it should be avoided the introduction of noise and overfitting, due to the increase of data dimensionality.

Aiming to add new meaningful features based on (RFDs), we defined a new function, named tuple Frequency Account in Validation (FAV), which permits to account for the number of times a tuple is involved in the validation of an (RFD) when it is coupled with other tuples. A more formal definition of the FAV function is provided below.

Definition 1 (Tuple Frequency Account in Validation (FAV)). *Given a relational database schema $\mathcal{R}$, defined on a set of attributes $attr(\mathcal{R}) = \{A_1, \dots A_m\}$, an instance r of $\mathcal{R}$ with n tuples, an (RFD) $\varphi : X_{\Phi_1} \xrightarrow{\Psi \leq \varepsilon} Y_{\Phi_2}$ holding on r, and a tuple t_i, the tuple frequency in validating*

φ can be defined as:

$$f_\varphi(t_i) = \frac{\sum_{j=1}^{n} \models_\varphi (t_i, t_j)}{n-1} \tag{1}$$

where $\models_\varphi (t_i, t_j)$ is a boolean function defined by the following formula:

$$\models_\varphi (t_i, t_j) = \begin{cases} 1, & \text{if } (t_i, t_j) \text{ satisfies } \varphi, \text{ with } i \neq j \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

In other words, $f_\varphi(t_i)$ counts the number of tuples that satisfy φ when compared with t_i.

The FAV function distributes the validation of an (RFD) throughout the tuples of the dataset. In this way, it is possible to characterize how much each sample (a tuple) is involved in the validation of an (RFD) φ, while maintaining the semantics of φ preserved. The proposed feature engineering methodology exploits the FAV function to add new features related to the discovered (RFDs) to the account dataset. Consequently, an (RFD) characterizing fake accounts should yield higher FAV values for fake account tuples; vice versa, it is expected that such accounts should present low FAV values for (RFDs) that do not characterize fake accounts. For more details about (RFD) concepts, refer to [9].

2.1. Ranking and Filtering FAV-based Features

Since the training set will necessarily include tuples of both fake and real accounts, the defined metrics consider both i) the class of each tuple, named *tuple type*, and ii) the category of accounts from which a (RFD) has been discovered, named (RFD) *type*. Then, according to them, it is possible to evaluate the expected FAV values. More specifically, we expected that a (RFD) holding on fake accounts only should provide FAV values that are high for fake accounts and low for real ones. Vice versa, a (RFD) holding on real accounts only should provide FAV values that are low for fake accounts and high for real ones. For this reason, we can state that the most significant FAV-based features are those that show the extremes of this behavior, by assigning a value of 1 when the *tuple type* and the (RFD) *type* match, and a value of 0 when the types do not match. However, having this kind of behavior is unrealistic, since it would define a perfect classification criterion for assigning an account to its proper category.

By following the previous considerations, we defined novel metrics, named FAV-based feature Evaluation Metrics (FEM), which allow the evaluation of the meaningfulness of FAV-based features in order to define a ranking and filtering strategy devoted to the minimization of the number of the newly added features in the classification models. A more formal definition of FEM metrics is provided below.

Definition 2 (FAV-based feature Evaluation Metrics (FEM)). *Given a relational database schema $\mathcal{R}$, defined on a set of attributes $attr(\mathcal{R}) = \{A_1, \dots A_m\}$, an instance r of $\mathcal{R}$ with n tuples, an (RFD) $\varphi : X_{\Phi_1} \xrightarrow{\Psi \leq \varepsilon} Y_{\Phi_2}$ holding on r, and the FAV-based feature generated from it f_φ, the evaluation of the meaningfulness of f_φ in discriminating account types can be defined as:*

$$\chi_\varphi = \frac{\sum_{t_i \in r} |e_i - f_\varphi(t_i)|}{n} \tag{3}$$

where e_i represents the expected value obtained according to the (RFD) type and the tuple type.

In other words, the proposed FEM metrics provide a value in the range $[0, 1]$ representing the meaningfulness that can be associated to a FAV-based feature. More specifically, for each tuple t_i of the training set, it measures how much the value $f_\varphi(t_i)$ differs from the corresponding expected value.

Notice that, the χ_φ metrics can be used for both ranking the FAV-based features and for filtering them when it is combined with an input threshold ε to form a selection constraint. In the latter case, only the FAV-based features satisfying the constraint $\chi_\varphi \leq \varepsilon$ will be then used during the classification process.

3. Experimental Evaluation

The experimental session started with the definition of the dataset to be considered in our evaluation. In particular, we have merged fake, verified, and real account datasets described in [6], and, for each of them, we have added an additional feature representing the type of each account. Starting from this mixed dataset, we have first encoded the categorical data into numerical ones by exploiting a Label Encoder approach [10], and then we have extended the number of features according to the proposed feature engineering strategy. The latter has been implemented considering the FAV value of the (RFDs) discovered through the DiMε algorithm [11], which has been set with an extent threshold equal to 0.5 and different attribute comparison thresholds, i.e., $Thrs$: 0, 1, 2, 3, 4, 8, and 12. These configurations allowed us to consider (RFDs) that might also be valid for a subset of accounts.

According to the resulting (RFDs), for each comparison threshold we constructed two datasets. The first one has been computed by adding the new FAV-based features as explained above, whereas the second one by only using the new FAV-based features (i.e., by removing the original features).

Each new dataset has been randomly split into training and test datasets with a proportion of 80% and 20%, respectively, and the effectiveness of supervised classification models has been evaluated in terms of *precision* (*P*), *recall* (*R*), and *accuracy* (*A*). Thus, we analyzed how the classification scores vary between the original dataset (named *Baseline*), the one augmented with the new features, and the one containing only the FAV-based features.

The experimental evaluation has involved Decision Tree [12], Random Forest [13], Support Vector Classification (SVC) [14], and Logistic Regression [15] as supervised classification models, by considering their versions available in the Scikit-learn[1] python library.

Experimental results of each classifier over the different configurations are shown in Figure 1. In particular, it is possible to notice that rows show the used classifiers, whereas columns show evaluation metrics. Additionally, each plot in Figure 1 highlights how the performances change when using original features augmented with FAV values (denoted by $16 + RFD$ in Figure 1), or FAV-based features only (denoted by RFD in Figure 1), by considering different comparison thresholds (denoted by RFD_i in each plot). Moreover, we also compared performances achieved on these datasets w.r.t. those achieved on the original dataset (denoted by, $Baseline$ in Figure 1).

In what follows, we discuss how the application of the proposed feature engineering strategy affects the performances of the trained classification models.

[1] https://scikit-learn.org/

Decision tree By applying the DT model in the context of fake account discrimination (see Figure 1) it is possible to notice that DT outperforms the baseline for each used evaluation metric when using original features combined with FAV values, as well as FAV values only, with all comparison thresholds. Specifically, the proposed feature engineering strategy enhances the capabilities of the DT model in the discrimination task because most of the added features have been selected in the tree construction, since they permit to infer more discriminative patterns with respect to those achievable with the baseline features. Furthermore, it is possible to notice that the DT model achieves the same results with all evaluation metrics when trained with the $16 + RFD$ or the RFD feature set w.r.t. all comparison thresholds. In detail, by performing further analysis, we observed that the model only selects FAV-based features to build the DT structure, considering the original features not beneficial for the training phase.

Random forest By applying the RF model in the context of fake account discrimination (see Figure 1), it is possible to notice that it outperforms the baseline for each evaluation metric, except on the recall, when using original features combined with FAV values, or the FAV values only, with all comparison thresholds. Moreover, it is possible to notice that the RF model achieves the same results for all evaluation metrics when trained on the $16 + RFD$ or RFD feature set with all comparison thresholds, except on the recall metric that does not present variations. This is what we expected, having observed that the proposed feature engineering strategy enhances the capabilities of the DT model in discriminating fake accounts, hence also RF indirectly benefits from it.

Support Vector Classification By applying the SVC model in the context of fake account discrimination we observed that it achieves the best results for each evaluation metrics, with all comparison thresholds, when trained on the RFD feature set, except for the recall metric that presents a slight decrease (see Figure 1). In particular, by performing a thorough analysis of fitting problems, we found that when trained with original or $16 + RFD$ feature set, such a model undergoes overfitting (with kernel set to Radial Basis Function - RBF) and underfitting (with kernel set to Sigmoid) phenomena. This is probably due to the fact that the original features do not permit to compute a hyperplane capable of discriminating accounts. Instead, using FAV values only implicitly guarantees the exploitation of semantic properties that permit a better discrimination capability when the SVC model is employed.

Logistic regression By applying the LR model in the context of fake account discrimination (see Figure 1), it is possible to notice that LR outperforms the baseline for each evaluation metrics when exploiting original features combined with FAV values, or FAV values only, with all comparison thresholds. In general, we have observed that our feature engineering strategy helps the LR model in the discrimination task, since most of the added features have either positive or negative weights, hence affecting the classification process.

K-nearest neighbors By applying the KNN model in the context of fake account discrimination (see Figure 1), it is possible to notice that KNN outperforms the baseline for each evaluation metrics when exploiting the FAV values only, with all comparison thresholds. On the other hand, when trained on the $16 + RFD$ feature set, no improvement w.r.t. the baseline is reported, for each evaluation metrics, except for the recall. In general, since the KNN model classifies a new account by comparing it through a distance metric with each training account, good results for

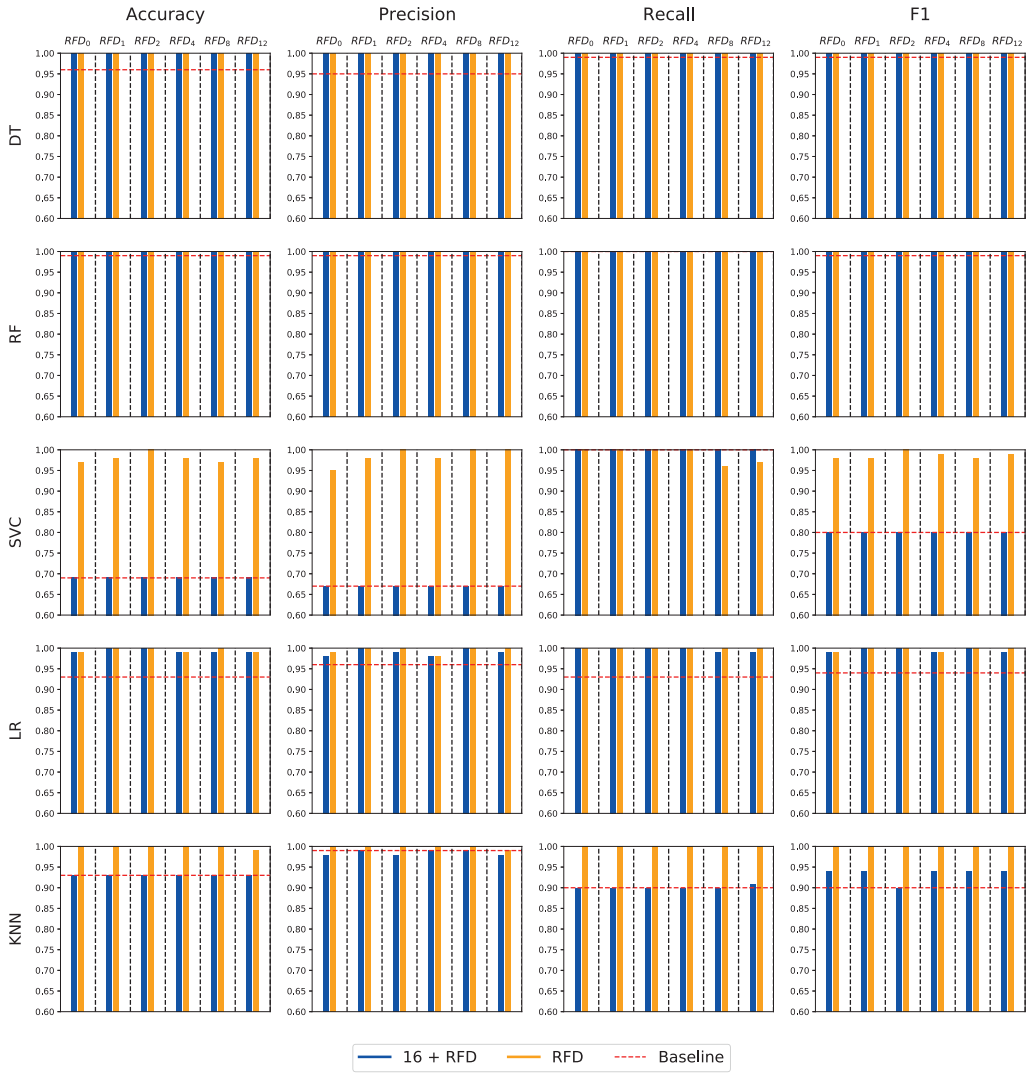

Figure 1: Evaluation metrics before and after applying the future engineering strategy.

each evaluation metrics highlight that the proposed feature engineering strategy produces FAV values having a strong correlation among accounts of the same class.

3.1. Feature Engineering Evaluation

In this section, we investigate the impact of filtering relevant FAV-based features in order to decrease the training time of classification models, while maintaining high performances in terms of fake account discrimination. In particular, Figure 2 reports accuracy (Accuracy), number of selected features (#Features), and training time (Time (s), expressed in seconds) by varying the comparison and the feature selection thresholds for each classification model.

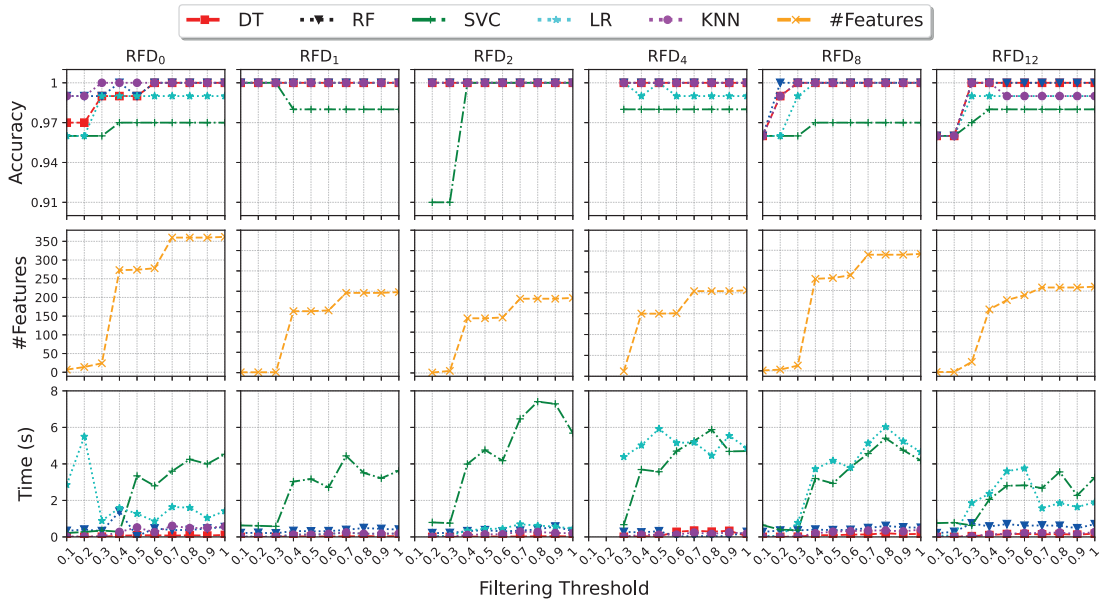

Figure 2: Trade-offs among accuracy, training time, and number of selected features by varying comparison and selection thresholds.

Moreover, feature selection thresholds reported on the x-axis are shared among all row plots and their maximum value (1) represents the selection of all FAV-based features. Instead, the other values represent a specific threshold to filter no relevant features w.r.t. comparison threshold. As can be seen in Figure 2, first-row plots show that the accuracy is maintained relatively high even if a significant number of features is removed. In particular, we achieve best results when considering the RFD_1 as a comparison threshold, which preserves the accuracy trend of all models and considers a restricted number of FAV-based features in the training set (see second-row plots in Figure 2). Additionally, third-row plots illustrate that the RFD_1 presents the shortest time recorded for the training phase of each classification model. In conclusion, we can observe that the combination of RFD_1 and 0.3 as comparison and selection thresholds, respectively, provide the best trade-off in terms of accuracy, training time, and number of features involved in the training phase. It is important to notice that, a more detailed discussion and further evaluations of the proposed feature engineering approach are shown in [6], in which we discuss the effectiveness of the proposed approach with respect to other feature engineering strategies proposed in the literature.

4. Conclusion

This work presents a new feature engineering strategy that exploits relaxed functional dependencies (RFDs) to enhance the capability of existing machine learning strategies in discriminating fake accounts. Evaluation results achieved over different machine learning models demonstrated that not only the proposed strategy permits to improve classification performances, but it never

negatively affects the application of models.

In the future, other than planning similar studies on different social network platforms, we would like to exploit other types of data dependencies, like graph dependencies, since they can potentially detect additional useful behavioral models to help discriminating fake accounts [16].

Acknowledgments

This Publication was produced with the co-funding of the European union - Next Generation EU: NRRP Initiative, Mission 4, Component 2, Investment 1.3 – Partnerships extended to universities, research centers, companies and research D.D. MUR n. 341 del 5.03.2022 – Next Generation EU (PE0000014 - "Security and Rights In the CyberSpace - SERICS" - CUP: H93C22000620001).

References

[1] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, M. Tesconi, Social fingerprinting: detection of spambot groups through dna-inspired behavioral modeling, IEEE Transactions on Dependable and Secure Computing 15 (2018) 561–576.

[2] Y. Liu, B. Wu, B. Wang, G. Li, Sdhm: A hybrid model for spammer detection in weibo, in: IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), IEEE, Beijing, China, 2014, pp. 942–947.

[3] Z. Chu, S. Gianvecchio, H. Wang, S. Jajodia, Detecting automation of twitter accounts: Are you a human, bot, or cyborg?, IEEE Transactions on Dependable and Secure Computing 9 (2012) 811–824.

[4] G. F. Campos, G. M. Tavares, R. A. Igawa, R. C. Guido, et al., Detection of human, legitimate bot, and malicious bot in online social networks based on wavelets, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 14 (2018) 26.

[5] C. Yang, R. Harkreader, G. Gu, Empirical evaluation and new design for fighting evolving twitter spammers, IEEE Transactions on Information Forensics and Security 8 (2013) 1280–1293.

[6] L. Caruccio, G. Cimino, S. Cirillo, D. Desiato, G. Polese, G. Tortora, Malicious account identification in social network platforms, ACM Transactions on Internet Technology 23 (2023) 1–25.

[7] M. Verleysen, D. François, The curse of dimensionality in data mining and time series prediction, in: International work-conference on artificial neural networks, Springer, Warsaw, Poland, 2005, pp. 758–770.

[8] S. B. Kotsiantis, D. Kanellopoulos, P. E. Pintelas, Data preprocessing for supervised leaning, International journal of computer science 1 (2006) 111–117.

[9] L. Caruccio, D. Desiato, G. Polese, Fake account identification in social networks, in: 2018 IEEE international conference on big data (big data), IEEE, 2018, pp. 5078–5085.

[10] R. Wang, R. Ridley, W. Qu, X. Dai, et al., A novel reasoning mechanism for multi-label text classification, Information Processing & Management 58 (2021) 102441.

[11] L. Caruccio, V. Deufemia, G. Polese, Mining relaxed functional dependencies from data, Data Min. Knowl. Discov. 34 (2020) 443–477.

[12] P. H. Swain, H. Hauska, The decision tree classifier: Design and potential, IEEE Transactions on Geoscience Electronics 15 (1977) 142–147.
[13] M. Pal, Random forest classifier for remote sensing classification, International journal of remote sensing 26 (2005) 217–222.
[14] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, K. R. Murthy, A fast iterative nearest point algorithm for support vector machine classifier design, IEEE transactions on neural networks 11 (2000) 124–136.
[15] F. O. Redelico, F. Traversaro, M. d. C. García, W. Silva, O. A. Rosso, M. Risk, Classification of normal and pre-ictal eeg signals using permutation entropies and a generalized linear model as a classifier, Entropy 19 (2017) 72.
[16] W. Fan, Y. Wu, J. Xu, Functional dependencies for graphs, in: Proceedings of the 2016 International Conference on Management of Data, ACM, San Francisco, California, USA, 2016, pp. 1843–1857.