UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE CIÊNCIAS
DEPARTAMENTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

RAIMUNDO TALES BENIGNO ROCHA MATOS

FEATURE SELECTION WITH LOW CORRELATED BINARY FEATURES FOR POTENTIAL TAX FRAUDSTERS CLASSIFICATION

FORTALEZA
2019

RAIMUNDO TALES BENIGNO ROCHA MATOS

FEATURE SELECTION WITH LOW CORRELATED BINARY FEATURES FOR POTENTIAL TAX FRAUDSTERS CLASSIFICATION

Tese apresentada à Coordenação do Programa de Pós-graduação em Ciência da Computação da Universidade Federal do Ceará como parte dos requisitos para obtenção do grau de Doutor em Ciência da Computação.
Orientador: Prof. Dr. José Maria Monteiro da Silva Filho

Co-Orientador: Prof. Dr. José Antônio Fernandes de Macêdo

FORTALEZA
2019

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

M382f Matos, Raimundo Tales Benigno Rocha.
Feature Selection with Low Correlated Binary Features for Potential Tax Fraudsters Classification / Raimundo Tales Benigno Rocha Matos. – 2019.
75 f.

Tese (doutorado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Ciência da Computação , Fortaleza, 2019.
Orientação: Prof. Dr. José Maria Monteiro da Silva Filho.
Coorientação: Prof. Dr. José Antônio Fernandes de Macêdo.

1. Feature Selection. 2. Low Correlated Features. 3. Tax Fraud Detection. 4. Association Rules. 5. Graph Centrality Measure. I. Título.

CDD 005

RAIMUNDO TALES BENIGNO ROCHA MATOS

FEATURE SELECTION WITH LOW CORRELATED BINARY FEATURES FOR POTENTIAL TAX FRAUDSTERS CLASSIFICATION

Tese apresentada à Coordenação do Programa de Pós-graduação em Ciência da Computação da Universidade Federal do Ceará como parte dos requisitos para obtenção do grau de Doutor em Ciência da Computação.

Aprovada em: Jun 19, 2019

BANCA EXAMINADORA

Prof. Dr. José Maria Monteiro da Silva Filho (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. José Antônio Fernandes de Macêdo (Co-Orientador)
Universidade Federal do Ceará (UFC)

Dr.ª Chiara Renso
Istituto di Scienza e Tecnologie dell'Informazione (ISTI/CNR - Itália)

Dr. Franco Maria Nardini
Istituto di Scienza e Tecnologie dell'Informazione (ISTI/CNR - Itália)

Prof. Dr. Cesar Lincoln Cavalcante Mattos
Universidade Federal do Ceará (UFC)

To my wife and daughters and to my parents.

ACKNOWLEDGEMENTS

One challenge as great as writing this thesis is to dedicate so little space to the people who were part of my career in the PhD in Computer Science at the Federal University of Ceará.

Thanks are due to my parents, who have made every effort to educate their children. Thank you my Father Raimundo Rocha Matos (*in memorian*) and my Mother Mrs. Maria Iondy Benigno Matos for dedication, encouragement and love. I'm very happy about it. The trajectory of my Father as a University Professor and my childhood playing in all the spaces of the Escola Superior de Agronomia de Mossoró certainly led me to the dream of studying a doctorate and becoming a researcher.

To my beloved daughters Alicia and Livia for their unconditional love and for the simple happiness in having them as daughters. I hope this accomplished work is seen as an example of the dedication of a Father who desires the best for his family. Love you. Thank you for being my daughters.

To my aunts, Ione (In memoriam) and Manize, whom I love as if they were my mothers. Thanks for the example, kindness and support when I came to live in Fortaleza and start my academic life. Your simplicity and love have laid the foundation for my growth.

To my brothers and nephews, who in the moments of my absence dedicated to study, have always understood that the future is made from constant dedication in the present!

To my in-laws, Mr. Melo (in memoriam) and Mrs. Graça for unconditional support with my daughters and for the family example.

To Prof. Dr. José Maria da Silva Monteiro Filho for guiding me in my doctoral thesis and also for his friendship and encouragement. You are a professional and a human being reference. Thank you for the constant demonstrations of wisdom and humility.

To Prof. Dr. José Antonio F. de Macedo, coordinator of the PhD in Computer Science at the UFC, for co-guiding my research in this thesis and for his patience and support during this journey. I had the opportunity to meet few professionals so dedicated and so brilliant. Always willing to help and teach, he makes no effort to contribute to the growth of everyone around him. Truly an example to follow.

To Dr. Francesco Lettich, for all his support and professionalism in the review and correction of my research. Serious and dedicated professional, taught me a lot about the importance of quality research.

Thanks also to Dr. Chiara Renso, Dr. Franco Maria Nardini and Dr. Raffaele Perego, from ISTI - CNR in Pisa, Italy, where I carried out part of my research and had the opportunity to evolve my knowledge and know the strength of scientific research conducted by a cohesive and high-level team. To the colleagues with whom I have lived daily in Italy and who have also helped in this journey: Vinicius Monteiro, Roberto Trani and Salvatore Trani.

I extend my thanks to the staff of MDCC-UFC, as well as to PhD fellows. Thanks for the support and teachings.

I thank all the teachers for providing me with not only rational knowledge, but the manifestation of the character and affectivity of education in the process of professional formation, not only for having taught me, but for having made me learn.

To the friends who rule to understand my absence in the moments of leisure and work.

I thank GOD for being able to name all these people at this very important moment. Thank you for putting them so capriciously into my life.

My most special and profound thanks could only be given to one person: my wife, Naukiana. Completing a doctorate is an epic journey of enormous learning. However, as great as the knowledge gained is the difficulty in completing it. My wife's support, love, and dedication were crucial in every step. A real family begins and stays in love, but consolidates itself in adversity. Saint Augustine said, "Love resists adversity, shows prudence in prosperity, is strong in suffering, rejoices in good news." Well, that's our family. Thank you, my inseparable companion.

> “Those who fall in love with practice without science are like a sailor who enters a ship without a helm or a compass, and who never can be certain whither he is going.”
>
> (Leonardo da Vinci)

RESUMO

Os métodos de Feature Selection fornecem uma maneira de reduzir o tempo de computação, melhorar o desempenho da classificação e um melhor entendimento dos dados em aplicações de aprendizado de máquina ou reconhecimento de padrões. Feature Selection tornou-se o foco de muita pesquisa em áreas aplicadas. Neste trabalho, usamos esta técnica para detectar possíveis fraudadores de impostos. A classificação de possíveis fraudadores a partir dos dados do contribuinte, com indicadores de fraude (features) binária, apresenta vários desafios: em primeiro lugar, estes dados costumam ter features com baixa correlação linear entre si. Além disso, as fraudes fiscais podem se originar de esquemas ilícitos intricados, o que, por sua vez, requer a descoberta de relacionamentos não lineares entre várias features. Finalmente, no conjunto de features existentes em nossos experimentos, apenas um pequeno número delas mostra alguma correlação com a classe alvo. A evasão fiscal representa um dos principais obstáculos enfrentados pelas economias dos países em desenvolvimento. Grandes quantidades de informações dos contribuintes foram coletadas por agências fiscais, abrindo assim a possibilidade de criar novas técnicas capazes de combater a evasão fiscal de forma muito mais eficaz do que as abordagens tradicionais. Neste trabalho propomos ALICIA, um novo método de seleção de features baseado em regras de associação e lógica proposicional com uma medida de centralidade para grafos cuidadosamente elaborada que tenta enfrentar os desafios acima e, ao mesmo tempo, sendo independente de técnicas específicas de classificação. Nosso método, ALICIA, quer capturar a inter-relação intrínseca entre os recursos na detecção de fraude fiscal. A metodologia proposta está estruturada em três fases: em primeiro lugar, o ALICIA gera um conjunto de regras de associação relevantes a partir de um conjunto de indicadores de fraude (features). Posteriormente, o ALICIA constrói um grafo, onde cada nó representa um subconjunto de features que resultam nas regras de associação, enquanto as arestas representam relacionamentos de associação entre subconjuntos de features. Finalmente, o ALICIA determina as features mais relevantes aplicando uma nova medida de centralidade, a *Fraud Feature Topological Importance*, nos vértices do grafo. Realizamos uma extensa avaliação experimental para avaliar a validade de nossa proposta em quatro diferentes conjuntos de dados do mundo real, onde comparamos nossa solução com oito outros métodos de seleção de features. Os resultados mostram que o ALICIA atinge valores de F-measure de até 76,88% e supera consistentemente seus concorrentes.

Palavras-chave: Seleção de Atributos. Atributos de Baixa Correlação. Detecção de Fraude

Fiscal. Regras de Associação. Medida de Centralidade em Grafos. Aprendizagem de Máquina.

ABSTRACT

Feature selection methods provides us a way of reducing computation time, improving prediction performance, and a better understanding of the data in machine learning or pattern recognition applications. It has become the focus of much research in areas of application. In this work, we use feature selection to select the most relevant features in order to improve the binary classification of potential tax fraudsters. Classify possible fraudsters from taxpayer data, with binary features, presents several challenges: firstly, taxpayer data typically have features with low linear correlation between themselves. Also, tax frauds may originate from intricate illicit schemas, which in turn requires to uncover non-linear relationships between multiple fraud indicators (features). Finally, in the set of features existing in our experiments, only a small number of them show some correlation with the targeted class. Tax evasion represents one of the major obstacles faced by the economies of developing countries. Vast amounts of taxpayer information has been collected by fiscal agencies, thus opening up the possibility of devising novel techniques able to tackle fiscal evasion much more effectively than traditional approaches. In this work we propose ALICIA, a new feature selection method based on association rules and propositional logic with a carefully crafted graph centrality measure that attempts to tackle the above challenges while, at the same time, being agnostic to specific classification techniques. ALICIA wants to capture the intrinsic interrelation between the features in tax fraud detection. The proposed methodology is structured in three phases: firstly, ALICIA generates a set of relevant association rules from a set of fraud indicators (features). Subsequently ALICIA builds a graph, where each node represents a subset of features resulting in the association rules, while edges represent association relationships between subsets of features. Finally, ALICIA determines the most relevant features by applying a novel centrality measure, the *Feature Topological Importance*, on the vertices of the graph. We perform an extensive experimental evaluation to assess the validity of our proposal on four different real-world datasets, where we compare our solution with eight other feature selection methods. The results show that ALICIA achieves F-measure scores up to 76.88%, and consistently outperforms its competitors.

Keywords: Feature selection. Low Correlated Features. Tax Fraud Detection. Association Rules. Graph Centrality Measure. Machine Learning.

LIST OF FIGURES

LIST OF TABLES

LISTA DE ABREVIATURAS E SIGLAS

FN	False Negatives
FP	False Positives
FS	Feature Selection
ML	Machine Learning
PR	Precision-Recall curves
ROC	Receiver Operator Characteristic curves
TN	True Negatives
TP	True Positives

CONTENTS

1 INTRODUCTION

1.1 Motivation

In Brazil, one of the major problems in tax management is tax evasion [1]. The correct payment of taxes by taxpayers guarantees the State to keep the necessary investments for society. Without taxes, the state can not guarantee health, education, sanitation, transportation, infrastructure, among other services essential to the population.

Tax evasion represents one of the major obstacles faced by the economies of developing countries. In the recent past, several initiatives were undertaken to develop applications that collect and leverage taxpayer information (NFE, 2019 (Retrieved June 15, 2019); CTE, 2019 (Retrieved June 15, 2019); SPED, 2019 (Retrieved June 15, 2019); MDFE, 2019 (Retrieved June 15, 2019)) to predict tax fraudsters. The amount of frauds, however, is still very high – for instance, the level of tax evasion in Brazil proved to be about 90 billions US$ in 2018 [2]. These numbers show that the fight against fraud is a crucial aspect of any fiscal system. This scenario opens possibilities for devising novel algorithms that can be used to tackle fiscal evasion much more effectively than traditional approaches.

Several Finance Departments are investing in expert systems to aid in decision making. Among the decisions of a tax administration, one of the most important is undoubtedly knowing who to inspect or who needs greater tax control. Machine Learning (ML) is currently an extremely important process for all organizations that have large databases. Government organizations belong to this context, where increasing computerization makes it possible to use various ML techniques and tools. Currently, the Finance Departments of the States of Brazil receive a large volume of data on the operations of companies registered in their registries, as well as information from other sources, which store their databases.

ML is an area of Artificial Intelligence whose objective is the development of computational techniques on learning, as well as the construction of systems capable of acquiring knowledge automatically. A learning system is a computer program that makes decisions based on accumulated experiences through successful solution of past problems. Within ML there is inductive learning, which can be understood in three parts: unsupervised learning,

[1] Tax evasion is any act that, consciously or unconsciously, legally or illegally, leads to non-payment or less payment of the tax due.

[2] https://ibpt.com.br/noticia/2734/Brasil-deixou-de-arrecadar-R-345-bi-por-sonegacao-de-impostos (retrieved: June 15, 2019)

semi-supervised learning, and supervised learning. In this study, we used supervised learning.

Supervised learning is the process of automatically creating a classification model from a sample base (called a training base) that has an attribute (column) called a class attribute. There are two aspects to consider in this process: which property should be used to describe the concept and how to combine these properties. Once the model is created, it can be used to automatically predict the class of an unclassified set of examples. In chapter 2 we present these concepts in a formal way.

In this work we consider a real-world case study involving the Treasury Office of the State of Ceará (SEFAZ-CE, Brazil)[3], the agency in charge of supervising more than 300,000 active contributors in the state of Ceará, Brazil. SEFAZ-CE maintains a large dataset containing vast amounts of information; its enforcement team, however, struggles to perform through inspections on taxpayers accounts as the inspection process involves the evaluation of countless fraud indicators, thus requiring burdensome amounts of time and being potentially prone to human errors.

With the application of ML techniques to these databases, the problem of focusing on the most relevant information has become very important. Thus, one of the main problems in ML is the selection of relevant features. There are several reasons for performing attribute selection. One of these reasons is that most computationally viable ML algorithms do not work well in the presence of a large number of features, i.e. Feature Selection (FS) can improve the accuracy of the classifiers generated by these algorithms. Another reason is that the selection of features improves the human's ability to understand the data and also, for example, the rules of induction generated by symbolic ML algorithms. A third reason for the FS is the high cost to acquire the information, since in many domains the collection of data can be very expensive. Finally, attribute selection can reduce the processing costs of large amounts of data.

In general terms, FS can be described in terms of a process that selects a subset of *relevant* features, among those available, to be subsequently used to build learning models. Many existing feature selection techniques focus on selecting feature subsets that are strongly correlated with some class; in the context of tax fraud detection, however, there is a need to find out and leverage relationships between features, as frauds may originate from complex, illicit schemes.

As we will see throughout this thesis, classification methods to detect potential

[3] http://www.sefaz.ce.gov.br/

tax fraudsters, without the use of feature selection, did not achieve results better than 58% of F-measure and the use of traditional FS methods didn't reach 70% of F-measure. Selecting an effective and more representative feature set is the subject of this work.

One common practice of filter type methods is to simply select the top-ranked fraud indicators (features) based on a linear correlation. A deficiency of this simple ranking approach, is that in tax fraud domain we have low or non correlated features among themselves and only a small number of these features show some correlation with the targeted class, but these are not good enough to perform a satisfactory classification. This happens, because some features insert noise to the classification task decreasing accuracy.

In the next section, we will examine the issues that guide the need for this scientific research.

1.2 Research Problems

Several data mining techniques have been extensively used to detect financial frauds (RAVISANKAR *et al.*, 2011; GLANCY; YADAV, 2011; KIRKOS *et al.*, 2007), insurance frauds (NGAI *et al.*, 2011), credit card frauds (BHATTACHARYYA *et al.*, 2011; SÁNCHEZ *et al.*, 2009), fraudulent transactions (LI *et al.*, 2012; PHUA *et al.*, 2010), and e-commerce frauds (ZHANG *et al.*, 2013; KIM *et al.*, 2013; RICHHARIYA *et al.*, 2012); these approaches, however, do not exploit*feature selection* to reduce the number of features considered: indeed, considered the sheer amount of indicators that can be used to characterize fraudsters, feature selection represents a powerful pre-processing tool that provides the potential to reduce the cost of feature measurement, speed up the testing process, and increase the efficiency and accuracy of classifiers (KIRA; RENDELL, 1992b; KOHAVI; JOHN, 1997).

The main goal of this thesis is to introduce a novel Feature Selection method in the context of tax fraudster detection. In this work we address some challenges not considered in the available literature:

1. Tax fraud indicators (features) have low or non-linear correlation between themselves. The ability to capture these relationships can improve the performance of feature selection methods.
2. Typically, only a small number of these features show some correlation with the targeted class.
3. All features and Class are binary.

Accordingly, we focus on the following research question:

- **RQ. How do we select the most appropriate subset of binary features, with low or non-linear correlation between themselves and possibly non informative for the target class, to improve potential tax fraudsters classification?**

In the next section, we discuss the main contribution of this thesis based on the challenges and question presented.

1.3 Thesis Contribution

The main contributions of this thesis, aligned to our research question, are the following:

- We analyze the behavioral pattern of fraud indicators, including the existence of correlation between them, as well as which are the most relevant fraud indicators and how can we measure the risk of a taxpayer committing a fraud. We proposed a method for classifying taxpayers in order to help detecting potential fraudsters and analyzed the feasibility of dimensionality reduction techniques as a way to create a scale of risk for frauds. The results were published in (MATOS *et al.*, 2015).
- We proposed a new method to determine the risk that taxpayers have to commit tax frauds. The method extends and improves (MATOS *et al.*, 2015) in that it uses a new technique to discover key indicators of frauds by resorting to the centrality measure over a graph of indicators: this implicitly captures relationships between key fraud indicators. The results were published in (MATOS *et al.*, 2017).
- In this thesis we propose a novel feature selection ranking method, ALICIA. The key idea behind ALICIA is to combine association rules and propositional logic with graph centrality measure to capture the hidden relationship between tax fraud features. In other words, ALICIA wants to leverage association rules, with a carefully crafted graph centrality measure, to identify the most relevant features for tax fraudsters classification. To achieve this, ALICIA is structured in three phases: in the first phase the approach generates a set of relevant fraud association rules from tax fraud data, where contributors are associated with a set of feature indicators. Still in this phase, we introduce an axiomatic description of dependency structures in association rules, based on propositional logic to improve the number of edges to second phase. Subsequently ALICIA builds a graph, where nodes represent subsets of features present in the association rules, while edges represent association relationships between the subsets. Finally, ALICIA evaluates the importance associated with the nodes of the graph according to a novel graph centrality measure, and returns a relevance ranking of the features.

To assess the quality of our proposal we conduct a performance comparison with eight, well-established feature selection methods, i.e., Information Gain (IG), Gain Ratio (GR), Correlation-based Feature Selection (CFS), Feature Selection via Eigenvector Centrality(ECFS), Relief-F, Fisher, Gradient boosted feature selection (GBFS) and XGBoost Feature Importance

(XGB-FI). We use three different real-world tax-payer datasets provided by SEFAZ-CE covering a period of time comprised between 2009 and 2011. The results show that ALICIA, coupled with an SVM-based classifier, achieves F-measure scores up to 76.88%, consistently outperforming its competitors, thus producing consistent results over time. A fourth dataset is used to analyze how well our strategy is able to generalize in fraud domain. ALICIA, in this case, using an XGB classifier also proved to be the best feature selection strategy when comparing with its competitors.

1.4 Thesis Organization

The remainder of this thesis is organized as follows: Chapter 2 provides some background on the key topics underlying this work, i.e., feature selection, association rule learning, and centrality measure. Chapter 3 presents our proposal, ALICIA. Chapters 4 and 5 present, respectively, the experimental setting and the experimental evaluation. Finally, Chapter 6 draws the final conclusions.

2 RELATED WORK AND BACKGROUND

With the aim to respond to our Research Question **RQ** (Section 1.2), this chapter presents the works related to this thesis and a background knowledge that supports our solution. First, we present works concerning on feature selection methods applied to fraud detection in Section 2.1. Then, we give basic foundations in *Feature Selection* in Section 2.3. This is important to understand how to select the most appropriate subset of features. Then, in order to understand how to capture the non-linear correlation between features, Section 2.4 provides an overview on *Association Rule Learning*, while Section 2.5 provides an overview on *Node Centrality* measures.

2.1 Feature selection methods applied to fraud detection

Fawcett e Provost (1997) present a technique to detect frauds involving cellular cloning. In their work the authors employs a single layer perceptron network coupled with a feature selection technique, called *sequential forward selection process* (KITTLER, 1986). In the experimental evaluation the authors show that their feature selection method allows to reduce the number of features from 198 to 11, while allowing the network to achieve 92% of accuracy; the authors finally report non-negligible misclassification costs.

Yang e Hwang (2006) propose an approach to detect frauds perpetrated in the health-care domain. The approach works by building a detection model from clinical pathways, where features are based on frequent control-flow patterns inferred from two datasets – one containing fraudulent instances while the other one containing regular instances. They then use a probability framework based on the Markov blanket filter to perform feature selection, which allows to reduce the number of features from 30,701 to 3,120. The experimental evaluation shows that their approach is capable of identifying several fraudulent cases that cannot be detected by manually constructed detection models, and that the use of feature selection increases the accuracy from 40% to 69%.

Similarly, Li *et al.* (2008) provides a comprehensive survey of statistical methods used to detect frauds perpetrated in health-care. Here, the authors point out how feature selection plays a very important role in this domain, yet most of the literature omit discussing the details behind the engineering of the features due to legal and privacy concerns.

Kotsiantis *et al.* (2006) explore the ability of machine learning techniques to detect firms issuing fraudulent financial statements. In this work the authors employ a feature selection method that ranks the relevance of features according to a statistical measure called *ReliefF*. ReliefF operates by determining the relevance of each feature according to its ability in disambiguating similar samples, where the similarity is defined in terms of *proximity* within the feature space. In the experimental evaluation the authors show how ReliefF is able to reduce the set of considered features from 25 to 8 features. Also, the authors compare several machine learning techniques, i.e., C.45, RBF, Bayesian Networks, KNN, and SVM, and show that C4.5 achieves the best results by classifying correctly the 85.2% of fraud cases, the 93.3% of non-fraud cases, and the 91.2% of cases in the validation sets.

Belhadji *et al.* (2000) propose an approach to choose fraud features that are deemed the most significant in predicting the probability that a claim is fraudulent. The approach is structured in three phases: first, their approach takes in input a set of fraud features that are provided by selected domain experts. Secondly, for each feature the approach calculates the associated conditional probability. Finally, the approach leverages the Probit regression to determine the most significant features.

Table 1 presents a summary of the feature selection methods used in the references presented in this section, as well as data domain, number of features, metrics and results obtained.

The following section analyzes the behavioral dynamics of tax evasion, using social networks e centrality measures.

2.2 Social networks and centrality measures applied to the Behavioral dynamics of tax evasion

Andrei *et al.* (2014) develop a model to show that the network structure determines in which way tax behavior is transmitted in a society. The model utilizes their rules for taxpayer behavior and apprehension of tax evaders in order to test the effects of network topologies in the propagation of evasive behavior. They use centrality metrics for measuring and analyzing networks.

Hashimzade *et al.* (2015) investigate the behavioural and social aspects of tax

FS methods	Classification algorithm	Domain	# features	# reduced features	Metric	Classification result
FAWCETT; PROVOST, 1997	Neural network	cellular cloning	198	11	Accuracy	92%
YANG; HWANG, 2006	Markov blanket filter	health-care	30,701	3,12	Accuracy	69%
LI et al., 2008	Survey	health-care				
KOTSIANTIS et al., 2006	C4.5	financial fraud	25	8	F-measure	88.79%
BELHADJI et al., 2000	Probit model	insurance fraud	54	23	Accuracy	80%

Table 1 – Feature selection methods applied to fraud detection.

compliance. in this work the authors show that specific attitudes and beliefs are also endogenously formed via the structure of the agents' social networks which are used to communicate auditing and compliance information.

Borgatti *et al.* (2009) explains that networks, cliques and communities of taxpayers are the relevant categories for social interaction of taxpayers because they describe how groups of otherwise isolated individuals may be formed.

The study of Korobow *et al.* (2007) illustrates that especially group effects are important for an individual's decision whether to evade or not. They find that the existence of social networks diminishes compliance.

A deficiency of these techniques is that they do not consider that features could be correlated among themselves in a non-linear way.

In the following section the theoretical foundations necessary for the development of the following chapters are discussed.

2.3 Feature selection

When considering the problem of detecting tax frauds, using a high number of features does not necessarily translate into a high classification accuracy and may require too many computational resources. To this end, feature selection methods can be used as a preprocessing tool to select features that are *relevant* for the problem considered – this reduces the number of *irrelevant* features while maintaining a good classification accuracy. We note that feature selection methods are different than feature extraction algorithms (DEVIJVER; KITTLER, 1982), in that the latter create new features originating from the combination or transformation of the initial features.

Two qualities that characterize features are considered by feature selection methods, regardless of the specificities of individual approaches: *relevance* and *redundancy* (YU; LIU, 2004). A feature is said to be *relevant* if it cannot be removed from the set to which it belongs without affecting its original conditional class distribution. A feature is considered to be *redundant* if it is highly correlated with other features. We note that features that may be considered irrelevant when taken individually may be extremely relevant when picked up in combination with other features. In general, using a good feature selection method can increase the classification accuracy with respect to other strategies (KIRA; RENDELL, 1992b; KOHAVI; JOHN, 1997). Determining an optimal subset of relevant features is challenging, since there is always a trade-off between subset *minimality*, i.e., a subset with the smallest possible amount of features, and subset *suitability*, i.e., a subset that guarantees the highest possible accuracy. In general, a proper trade-off is domain-dependent. In light of the above considerations, we introduce the following problem statement.

Definition 2.3.1 *Given a feature set F, we define the problem of feature selection as the problem of finding out a near optimal subset of features $M \subseteq F$ among the competing $2^{|F|}$ candidate subsets.*

The definition of optimality may vary according to the specificities of the scenario considered. Also, using naïve (brute-force) approaches is unfeasible with most of the datasets, hence feature selection methods try to exploit some kind of heuristic to effectively reduce the search space (DASH; LIU, 1997; RUIZ *et al.*, 2005) and find near-optimal solutions.

Existing feature selection methods fall under three categories (GUYON; ELISSEEFF, 2003), i.e., *filter*, *wrapper* and *embedded*, where the categorization stems from the different strategies used to *measure* the optimality of subsets of features.

2.3.1 Filter methods

Filter methods use statistical properties characterizing features to *score* and *rank* subsets of features, independently of specific learning algorithms used (CHANDRASHEKAR; SAHIN, 2014; LIU; MOTODA, 2007). We note that some filter methods – for instance, those based on mutual information criteria – provide a generic approach to select features, hence they may be not suitable for some machine learning methods. Filtering methods can be employed as a preprocessing tool too, to reduce space dimensionality and avoid overfitting. Filter methods typically use all the training data to look for relevant subsets of features.

Entropy represents a commonly used measure that quantifies the information content of a source, and constitutes the foundation of many filter methods. To evaluate entropy, in this work we consider the following measures: Information Gain (IG) (LIU; MOTODA, 2012), Gain Ratio (GR) (LIU; MOTODA, 2012; KAREGOWDA *et al.*, 2010) and Correlation Feature Selection (CFS) (HALL, 1999).

Let us suppose that Y represents the discrete random variable modeling the occurrences of the classes in S. Then, the entropy of Y is defined as:

$$H(Y) = -\sum_{y \in Y} p(y)log(p(y)) \tag{2.1}$$

where $P(y)$ denotes the probability associated with the class y. Intuitively, the lower the entropy, the more the information in S is "predictable" – this, in turn, occurs whenever S contains samples whose classes have a low or high probability to occur.

Let us now suppose that the values of Y are partitioned according to the values that some feature can assume – we describe the occurrences of such feature by means of a discrete random variable X: if the conditional entropy of Y with respect to X, that is, the amount of information needed to describe the outcome of Y given that the value of X is known, is less than the entropy of Y, then there exists some *relationship* between Y and X. Such measure is denoted by $H(Y|X)$ and is defined as:

$$H(Y|X) = -\sum_{x \in X} p(x) \sum_{y \in Y} p(y|x)log(p(y|x)) \tag{2.2}$$

where $P(y|x)$ represents the posterior probability of $Y = y$ given the evidence of $X = x$. In the following we provide a brief overview on the measures considered in this work.

2.3.1.1 Information Gain

Information gain (IG) (LIU; MOTODA, 2012) measures the decrease in entropy when classes are partitioned according to the values some feature can assume, the higher the IG associated with X, the more the associated feature provides useful information about the classes.

Let us consider a dataset $S = \{s_1, \cdots, s_m\}$, where a generic sample $s \in S$ has the form $s = (x^{(1)}, \cdots, x^{(n)}, y)$, with $x^{(j)}$ representing the value assumed by the j-th feature in the sample and y representing the *label* associated with the sample. In this context we also assume that any label y can assume only a finite number of values (*classes*), i.e., $y \in C$, with C representing the set of classes. As such, IG is defined as follows:

$$IG(Y,X) = H(Y) - H(Y|X). \tag{2.3}$$

When applied to feature selection, features that achieve high IG are deemed to be the most relevant.

Finally, we report that IG's main weakness lies in its bias towards features featuring large numbers of distinct values.

2.3.1.2 Gain Ratio

The Gain Ratio (GR) represents a measure that leverages IG while attempting to tackle its bias limitation (LIU; MOTODA, 2012; KAREGOWDA *et al.*, 2010). It is defined as follows:

$$GR(Y,X) = \frac{IG(Y,X)}{H(X)} \tag{2.4}$$

From Eq.2.4, we see that GR attempts to predict the class Y by dividing IG with the entropy of X. The values produced by GR always fall in the range $[0,1]$. A value of $GR = 1$ indicates that the knowledge of X completely predicts Y, and $GR = 0$ means that there is no relation between Y and X. In opposition to Information Gain, the Gain Ratio favors feature with fewer values.

2.3.1.3 Correlation-based Feature Selection (CFS)

The Correlation-based Feature Selection (CFS) measure (HALL, 1999) uses a correlation-based heuristic to rank subsets of features.

More precisely, CFS evaluates subsets of features X according to the following hypothesis: *good feature subsets contain features that are highly correlated with some class Y, yet uncorrelated with each other.* To achieve this, CFS uses a *best-first-search heuristic*: first, it calculates a matrix of feature-class and feature-feature correlations from the training data; subsequently, it scores a subset V of k features in the following way:

$$Merit_{V_k}(S,Y) = \frac{k\overline{r_{cf}}}{\sqrt{k+k(k-1)\overline{r_{ff}}}} \tag{2.5}$$

Here, $\overline{r_{cf}}$ is the average feature-class correlation, while $\overline{r_{ff}}$ is the average feature-feature intercorrelation.

Symmetrical uncertainties (SU) are used in CFS to estimate the degree of association between discrete features or between features and classes. More specifically, it measures the intercorrelation between two features, or the correlation between a feature X and a class Y with binary labels. SU can be defined as follows:

$$SU(S,Y) = 2.0 * \left[\frac{GR(S,Y)}{H(X_i) + H(Y)} \right], \tag{2.6}$$

The subset yielding the highest merit is finally selected.

2.3.1.4 RELIEF and RELIEF-F

Similarly to the case of CFS, in (KIRA; RENDELL, 1992a) the authors observe that feature selection conducted by means of information gain or gain ratio assumes that attributes are independent to each other, thus ignoring possible relationships between features. To address this limitation, the authors propose an algorithm called RELIEF.

RELIEF revolves around the idea that "good" features should have the same value among instances belonging to the same class, while they should possess different values between instances having different class. More formally, let $W[f]$ be the score associated with some feature f; then, $W[f]$ is computed as follows:

$$\begin{aligned} W[f] = & P(\text{different value of } f \mid \text{nearest instance with different class}) - \\ & P(\text{different value of } f \mid \text{nearest instance with same class}), \end{aligned}$$

where P denotes the probability associated with an event. Here we note that the closer to one the score of a feature, the better the feature.

To compute the aforementioned probabilities, RELIEF operates as follows: first, the algorithm randomly extracts m instances from the dataset, where m is smaller or equal than the number of total instances – we note that the greater m, the more exact the scores associated with the features. Subsequently, for each of instance RELIEF finds out (i) the nearest instance having different class and (ii) the nearest instance having the same class; such instances are found out via a suitable *distance* function. We note that this serves the purpose of taking into account possible relationships between features. Once the two nearest instances are found, RELIEF proceeds to update the score of each feature, using again appropriate distance functions.

Several improvements were proposed over time with respect to the original algorithm. In this work we consider RELIEF-F (KONONENKO, 1994): among the various improvements, we report that this variant is able to cope with noisy or missing data, and is able to process multi-class datasets.

2.3.1.5 Fisher Score

The key idea of Fisher score (GU *et al.*, 2012) is to find a subset of features, such that in the data space spanned by the selected features, the distances between data points in different classes are as large as possible, while the distances between data points in the same class are as small as possible. It computes a score for each feature independently. In other word, it only considers $x^j \in \mathbb{R}^{1\times n}$. In this case, there are only $\binom{F}{1} = F$ candidates. Let n_i denote the number of data points in class i. Let μ_i^j and σ_i^j be the mean and standard deviation of ith class, corresponding to the jth feature. Let μ^j and σ^j denote the mean and standard deviation of the whole data set corresponding to the jth feature. Then the Fisher Score of the jth feature is computed as follows,

$$F(x^j) = \frac{\sum_{i=1}^{c} n_i(\mu_i^j - \mu^j)^2}{\sum_{i=1}^{c} n_i(\sigma_i^j)^2} \tag{2.7}$$

After computing the Fisher score for each feature, it selects the top-*m* ranked features with large scores. Because the score of each feature is computed independently, the features selected by the heuristic algorithm is suboptimal. The heuristic algorithm fails to select those features which have relatively low individual scores but a very high score when they are combined together as a whole.

2.3.1.6 Feature selection via eigenvector centrality (ECFS)

Feature selection via eigenvector centrality (ECFS) (ROFFO; MELZI, 2016), proposes a graph-based feature selection algorithm that ranks features according to the *graph eigenvector* centrality measure (BONACICH, 1987). The underlying idea is to map the problem to an affinity graph (where features are the nodes), and to model pairwise relationships among feature distributions by weighting the edges connecting them. It assigns a score of “importance” to each feature by taking into account all the other features mapped as nodes on the graph. The graph is weighted according Fisher criterion (equation 2.7) and mutual information (ZAFFALON; HUTTER, 2002):

$$m_i = \sum_{y\in Y} \sum_{z\in x^i} p(z,y) log(\frac{p(z,y)}{p(z)p(y)}), \tag{2.8}$$

Where Y is the set of class labels, and $p(\cdot,\cdot)$ is the joint probability distribution. Note that each feature distribution x^i is normalized so as to sum to 1. A kernel k is then obtained by the matrix product

$$k = (F \cdot m^\top),$$

Where F (equation 2.7) and m (equation 2.8) are $n \times 1$ column vectors normalized in the range 0 to 1, and k results in a $n \times n$ matrix.

In order to capture the amount of variation or dispersion of features from average, they use a second feature-evaluation metric based on standard deviation:

$$\Sigma(i,j) = max(\sigma^{(i)}, \sigma^{(j)}),$$

where σ being the standard deviation over the samples of X, and Σ turns out to be a $n \times n$ matrix with values $\in [0,1]$.

Finally, the adjacency matrix A of the graph G is given by:

$$A = \alpha k + (1-\alpha)\Sigma,$$

where α is a loading coefficient $\in [0,1]$. The generic entry a_{ij} accounts for how much discriminative are the feature i and j when they are jointly considered; at the same time, a_{ij} can be considered as a weight of the edge connecting the vertices i and j of a graph, where the ith vertex models the ith feature distribution.

After generating the adjacency matrix A of the graph G, this method uses Eigenvector Centrality (EC) to identifying the most important nodes within a graph (e.g., the relative importance of nodes). The basic idea behind the EC is to calculate v_0 the eigenvector of A associated to the largest eigenvalue. Its values are representative of how strongly each node is connected to the other nodes.

2.3.2 Wrapper methods

Wrapper methods explore the whole set of features, by means of a predictive model, to score feature subsets; these methods usually provide the best feature subsets, since they allow higher predictive performance than filter methods. Wrapper methods offer a simple yet powerful way to address the problem of feature selection, regardless of the choice of a specific machine learning approach – indeed, in this context any machine learning approach can be seen as a black box.

In its most general formulation, a wrapper method consists in using the prediction performance of a given machine learning algorithm to assess the relative usefulness of subsets of features.

In practice, one needs to define (i) how to search the space of all the possible subsets of features, (ii) how to assess the prediction performance of a machine learning algorithm to guide the search and halt it, and (iii) which predictor to use.

An exhaustive search can conceivably be performed if the number of features is not too large. However, the problem is known to be NP-hard (AMALDI; KANN, 1998), thus the search becomes computationally intractable as the amount of features gets larger.

As such, wrapper methods are often criticized because they tend to require massive amounts of computation – indeed, these methods are often considered *brute force* methods.

However, efficient search strategies can be exploited to reduce their computational cost and do not necessarily imply losses in prediction performance. All in all, whenever we use a machine learning algorithm as a black box wrapper methods prove to be remarkably universal, simple, and useful; unfortunately, they tend to be slow, since the induction algorithm is usually called repeatedly.

2.3.3 Embedded methods

Embedded methods perform feature selection during the training process and are usually tied to specific machine learning algorithms. For instance, decision trees such as CART (BREIMAN *et al.*, 1984) have a built-in mechanism to perform feature selection; other embedded methods guide their search by estimating changes in the objective function value that results when making moves in the feature subset space. All in all, embedded methods may be more efficient in several aspects than the methods reported above, i.e., they make better use of the data available by not splitting the training data into training and validation sets, and tend to reach a solution faster by avoiding to retrain a predictor from scratch each time a new feature subset is considered.

2.4 Association rule learning

The goal of association rule learning is to identify associations between data records (items) that are *related*. To achieve this, the key idea is to find subsets of items whose presence is correlated with the presence of some other item in the same transaction. Apriori (AGRAWAL; SRIKANT, 1994) represents an association rule technique that is widely used in market basket analysis, cross marketing and customer buying pattern. Association rule techniques represent promising tools in the context of fraud detection (SÁNCHEZ *et al.*, 2009; PHUA *et al.*, 2010; KIM *et al.*, 2013), since they can be used to discover hidden relationships among various fraud features.

2.4.1 Association rule

Let us suppose that $I = \{i_1, \ldots, i_m\}$ represents a set of *items*. Then, an *association rule* represents an implication of the form $X \Rightarrow Y$, where $X, Y \subseteq I$ and $X \cap Y = \emptyset$.

Association rules are incrementally derived from a set of transactions, D; their strength is measured in terms of *support* and *confidence* (AGRAWAL; SRIKANT, 1994). More specifically, let us suppose that $\sigma(Z)$ represents the *support of an itemset*, i.e., the number of transactions where the itemset Z appears, while $N = |D|$ represents the overall number of transactions in D. Then, *support* of an association rule is defined as:

$$supp(X \Rightarrow Y) = \frac{\sigma(X \cup Y)}{N} \tag{2.9}$$

The other important measure is the *confidence* of a rule, which determines how frequently the item(s) in Y appear(s) in instances that contain X:

$$conf(X \Rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)} \tag{2.10}$$

Finally, the *lift* of a rule represents a correlation measure between itemsets that can be used to augment the *support-confidence* framework (HAN; KAMBER, 2006):

$$lift(X \Rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X) \cdot \sigma(Y)} \tag{2.11}$$

In general, $lift(X \Rightarrow Y) < 1$ indicates that the occurrence of X is negatively correlated with the occurrence of Y, i.e., the lower the lift, the more the occurrence of X implies the occurrence of Y. $lift(X \Rightarrow Y) > 1$ represents the symmetric case. Finally, $lift(X \Rightarrow Y) = 1$ indicates the absence of correlation between X and Y, i.e., the two itemsets are independent.

2.5 Node centrality measures

The main goal of node centrality measures is to capture the *importance* that individual nodes have within a graph (FREEMAN, 1977). In general, central nodes are interesting since they have three characteristics that distinguish them from regular nodes: they possess more edges (relationships), their average distance with other nodes is lower than regular nodes, and they possess a major role in regulating the flow between nodes of the graph.

Considering the ubiquity of graphs, the concept of centrality is investigated in a relevant amount of domains (OPSAHL; PANZARASA, 2009; BARRAT *et al.*, 2004; DOREIAN *et al.*, 2005), including feature selection (MORADI; ROSTAMI, 2015; BONEV *et al.*, 2013; ZHANG *et al.*, 2011). Indeed, feature selection has been used in the past to rank single features. However, in most datasets features are not independent and combining them provides much more information than considering them individually. As such, feature subset selection can be conducted by exploiting some graph centrality measure to capture relationships between the nodes of the graph.

In the context of this work, we consider three different measures of centrality (FREEMAN, 1978): *degree* centrality, *closeness* centrality, and *betweenness* centrality.

2.5.1 Degree centrality

Let us suppose we have a graph $G = (V, E)$, where V represents the set of vertices and E represents the set of edges. Then, we define the *degree* centrality of a node $v \in V$, $D(v)$, as the number of nodes with which v is connected:

$$D(v) = \deg(v) \tag{2.12}$$

Intuitively, a node with high degree centrality may indicate that the node has a crucial role in a graph. This measure has a major limitation in that it does not take into account other properties of the graph. For instance, even if a node possesses a very high degree centrality, its average distance with respect to its neighbors may be very high; this, in turn, may represent an important factor to consider when we consider scenarios where some kind of resource must be retrieved as quickly as possible.

The measure of *closeness centrality* (SABIDUSSI, 1966) tries to overcome the aforementioned limitation by taking into account the distance.

2.5.2 Closeness centrality

Given a directed graph $G = (V,E)$, a vertex $v \in V$, and some distance function d, we define the closeness centrality of v as:

$$CC(v) = \frac{1}{\sum\limits_{u \in V, u \neq v} d(v,u)}, \tag{2.13}$$

where $d(v,u)$ represents the distance between v and some $u \in V$.

One of the main drawbacks of closeness centrality is the lack of applicability when considering networks that have disconnected components. Indeed, two nodes that belong to different components have infinite distance. The *betweenness centrality* measure (WIKIPEDIA, 2017) attempts to overcome this limitation. The underlying idea is to determine the amount of shortest paths that pass through a given node – as such, it measures the ability of a node to *funnel* flows within the network.

2.5.3 Betweenness centrality

Given a directed graph $G = (V,E)$ and a vertex $v \in V$, we define the betweenness centrality of v as:

$$BC(v) = \sum_{s,t \in V, s \neq t \neq v} \frac{\sigma_{s,t}(v)}{\sigma_{s,t}}, \tag{2.14}$$

where $\sigma_{s,t}$ represents the total number of shortest paths between some node $s \in V$ to some node $t \in V$, while $\sigma_{s,t}(v)$ represents the number of shortest paths between s and t that pass through v.

3 LEVERAGING FEATURE SELECTION TO DETECT POTENTIAL TAX FRAUDSTERS

In this section we introduce ALICIA, a new feature selection method that attempts to answer the research question introduced in Chapter 1: **How do we select the most appropriate subset of binary features, with low or non-linear correlation between themselves and possibly non informative for the target class, to improve potential tax fraudsters classification?**

The key idea behind ALICIA is to combine association rules and propositional logic with *graph centrality measure* to capture the hidden relationship – non-linear correlation – between tax fraud features. In other words, ALICIA wants to leverage *association rules*, with a carefully crafted graph centrality measure, to identify the most relevant features for tax fraudsters classification. Our method is structured in three phases – (Algorithm 1) provides the pseudocode.

In the first phase, GENERATERULES (line 2), ALICIA takes in input a data matrix *S*, i.e., a matrix where each row refers to a single taxpayer while each column is associated with a specific tax fraud feature, to generate the set of *relevant* fraud association rules, $rules$. In the second phase, GENERATEGRAPH (line 3), ALICIA builds from $rules$ the graph of association rules *G*, where each node represents a subset of tax fraud features present in any of the associative rules in *rules*, and each edge represents a specific association rule between a pair of subsets of features. In the third final phase, RANKFEATURES (line 4), ALICIA concludes by producing the relevance ranking (in descending order) of the nodes of *G* – to this end we leverage a novel centrality measure called *Feature Topological Importance* (FTI).

In the following we provide the details behind each phase.

3.1 Phase I – Generating the set of relevant fraud association rules

The goal of this phase is to produce a set of association rules that can be subsequently used to determine the final set of relevant features. Algorithm 2 reports its pseudocode.

GENERATERULES takes in input a data matrix *S*, where a generic element $s_{i,j} \in S$ indicates whether the *i*-th taxpayer is associated (1) or not (0) with the *j*-th tax fraud feature. The algorithm takes also in input the values used by the Apriori algorithm to determine the association rules, i.e., support (*supp*) and confidence (*conf*). Input variable $dependency$ is a boolean used to choose the use of dependency structures (subsection 3.1.1) to add more rules. GENERATERULES first computes from *S* the set of association rules, *APR*, by means of the Apriori algorithm (line 3). Finally, it returns the subset of rules $RARS \subseteq APR$ whose *Lift* is

Algoritmo 1: ALICIA

```
Input :
    • $S$, the data matrix.
    • $supp$, the support threshold to be used with Apriori.
    • $conf$, the confidence threshold to be used with Apriori.
    • $lift$, the lift threshold to be used with Apriori.
    • $n$, the n-reachability threshold used when ranking the features.
    • $maxFeatsVertex$, maximum number of features associated with a vertex.
    • $dependency$, boolean to indicate the use of dependency in rules generation
begin
    $rules \leftarrow$ GENERATERULES($S, supp, conf, lift, dependency$)        // Phase I
    $G \leftarrow$ GENERATEGRAPH($rules$)                                        // Phase II
    $KeyFeatures \leftarrow$ RANKFEATURES($G, n, maxNumFeatVertex$)             // Phase III
    return $KeyFeatures$
```

above the *lift* threshold (lines 4–5) and if choose by user, it increases the number of rules using dependency structures (line 6).

Technically, association refers to any relationship between two variables, whereas correlation is often used to refer only to a linear relationship between two variables.

3.1.1 Phase I.a – Dependency structures of association rules

In this phase, we introduce an axiomatic description of dependency structures in association rules. The axioms specify those families of dependencies "A determines B" which can hold for some association rule in *RARS*.

These dependencies allow our method ALICIA to create new rules, increasing the number of edges in the graph (subsection 3.2), in order to modify the value of the centrality of the nodes in the graph (subsection 3.3).

If α is a set of attributes, an ordered pair (A,B) with $A \subseteq \alpha$, $B \subseteq \alpha$ will be referred to as a dependency. $\varphi = \{(A,B) \mid A \subseteq \alpha, B \subseteq \alpha\}$ is the set of all dependencies over α.

Algoritmo 2: GENERATERULES

```
  Input :
      • $S$, the data matrix.
      • $supp$, the support threshold.
      • $conf$, the confidence threshold.
      • $lift$, the lift threshold.
      • $dependency$, boolean to indicate the use of dependency
1 begin
2     $RARS \leftarrow \emptyset$
3     $APR \leftarrow Apriori(S, supp, conf)$
4     foreach $ap \in APR$ do
5         if $(Lift(ap) \geq lift)$ then $RARS \leftarrow RARS \cup \{ap\}$
6     if $(dependency = true)$ then $RARS \leftarrow RARS \cup \{SetOfDependencies\}$
7     return $RARS$
```

If *AR* is an association rule, we say that the dependency structure of *AR* is the family:

$$\tau_{AR} = \{(A,B) \mid A \subseteq \alpha, B \subseteq \alpha, A \rightarrow_{AR} B\} \tag{3.1}$$

Using propositional logic (HEGEL, 2014; KRAJICEK *et al.*, 1995; FAGIN, 1977), we propose the following axioms:

3.1.1.1 Axiom of reflexivity

If A is a set of items and B is a subset of A, then A determines B.

$$\text{if } B \subseteq A \text{ , then } A \rightarrow B \tag{3.2}$$

3.1.1.2 Axiom of composition

If A determines B and A determines C, then A determines BC[1].

$$\text{if } A \rightarrow B \text{ and } A \rightarrow C \text{ , then } A \rightarrow BC \tag{3.3}$$

3.1.1.3 Axiom of transitivity

If A determines B and B determines C, then A determines C.

$$\text{if } A \rightarrow B \text{ and } B \rightarrow C \text{ , then } A \rightarrow C \tag{3.4}$$

Let τ be a subset of φ, and let $\rightarrow$ be defined by $A \rightarrow B$ iff $(A,B) \in \tau$. Then τ will be called a full family of dependencies over α if:

1. $(\tau 1)$ if $E \rightarrow F$ and $F \rightarrow G$ then $E \rightarrow G$ (transitivity);
2. $(\tau 2)$ if $E \rightarrow F$ and $E \rightarrow G$ then $E \rightarrow FG$ (composition);
3. $(\tau 3)$ if $E \rightarrow F,G$ then $E \rightarrow F$ and $E \rightarrow G$ (reflexivity, transitivity);

This phase is not mandatory and the dependency structure axioms $(\tau 1)$ to $(\tau 3)$ are not intended to be an optimal choice, however they do provide a try to change the importance ranking of the features (subsection 3.3) in some datasets, as demonstrated in Chapter 5.

[1] Let *BC* represent a compound item, the union of items *B* and *C*.

3.2 Phase II – Building the graph of association rules

In this phase ALICIA builds the graph of association rules G from the set of relevant association rules *rules* – the idea is to subsequently use G to rank the features. Algorithm 3 reports the pseudocode.

The algorithm begins by initializing the graph G, setting V and E to $\emptyset$ (line 2). Subsequently, each association rule $(X \rightarrow Y) \in rules$ is used to update G; more specifically, for each rule in *rules* the algorithm creates (if not already present) two *nodes*, representing respectively the subsets of features X and Y, and an *edge* (X,Y) that represents the association rule between the two. GENERATEGRAPH terminates by returning G.

3.3 Phase III – Feature ranking

The goal of this phase is to rank the original features starting from the information in G. To this end, we first need to introduce the new centrality measure used by ALICIA, i.e., the *Feature Topological Importance* (FTI) measure. It is a new method to asses the strength of indirect interactions between features, quantifying longer pathways to help determining which features have more direct or indirect effects on others.

Let $G = (V,E)$ a directed graph, and $i \in V$ some vertex in G: a node $v \in V$ is said to be *n-reachable* from i if the number of edges (hereinafter denoted by *number of steps* for simplicity) of the shortest path connecting i to v is *less or equal* than n. Let us also define $a_n(G,i,v)$ as the *influence* of i on any node $v \in V$ that is *n-reachable* from i: the value of $a_n(G,i,v)$ is determined as:

$$a_n(G,i,v) = 1/D_n(G,i), \tag{3.5}$$

where $D_n(G,i)$ represents the overall number of vertices that are n-reachable from i.

Algoritmo 3: GENERATEGRAPH

```
  Input  : $rules$, the set of relevant associative rules.
1 begin
2     $V \leftarrow \emptyset, E \leftarrow \emptyset$
3     foreach $(X \rightarrow Y) \in rules$ do
4         if $X \notin V$ then $V \leftarrow V \cup \{X\}$
5         if $Y \notin V$ then $V \leftarrow V \cup \{Y\}$
6         if $(X,Y) \notin E$ then $E \leftarrow E \cup (X,Y)$
7     return $G = (V,E)$
```

Example: $a_4(G,i,v) = 1/8$ implies that (i) i has $D_4(G,i) = 8$ vertices that are 4-reachable from it and that (ii) any node $v \in V$ that is 4-reachable from i is subjected to i's $a_4(G,i,v) = 1/8$ influence.

Finally, let us define $\sigma_n(G,i)$ as the *total n-step influence* of i as the sum of its influence on every node that is n-reachable from it. More formally:

$$\sigma_n(G,i) = \sum_{v \in V} a_n(G,i,v) \tag{3.6}$$

At this point we can define the *Feature Topological Importance* (FTI) centrality measure.

Definition 3.3.1 (Feature Topological Importance) *Let $G = (V,E)$ be the graph of association rules and $n \geq 1$ be an n-reachability threshold: then, given a vertex $v \in V$, v's Feature Topological Importance score is computed as follows:*

$$FTI(G,n,v) = \frac{\sum_{m=1}^{n} \sigma_m(G,v)}{\max(1,n-1)}. \tag{3.7}$$

The intuition behind FTI is to measure the importance of a node v by examining how "well" it is connected to other nodes in the graph, i.e., the more the nodes that are n-reachable from v, the more the importance attributed to v.

We now introduce the algorithm in charge of ranking the set of features, RANKFEATURES (Algorithm 4). RANKFEATURES takes in input the graph of association rules $G = (V,E)$, the variable n that represents the n-reachability criterion, and the variable *maxFeatsVertex* – the latter represents the maximum number of features a vertex may have associated in order to be considered during this phase. First, RANKFEATURES initializes *RankFeatures* to an empty set (line

Algoritmo 4: RANKFEATURES

```
Input:
    • $G = (V,E)$, the graph of association rules.
    • $n$, the n-reachability threshold.
    • $maxFeatsVertex$, the maximum number of features admitted in a vertex.
Output: The ranking of the features, $RankFeatures$
1  begin
2      $RankFeatures \leftarrow \emptyset$
3      $V' \leftarrow getVertices(V, maxFeatsVertex)$
4      foreach $v \in V'$ do
5          $SetFeaturesVertex \leftarrow getFeaturesVertex(v)$
6          $score \leftarrow \frac{FTI(G,n,v)}{|SetFeaturesVertex|}$
7          foreach $f \in SetFeaturesVertex$ do $RankFeatures \leftarrow update(RankFeatures, (f, score))$
8      Sort features in $RankFeatures$ according to the associated scores
9      return $RankFeatures$
```

2) and extracts from V the subset of vertices that are associated with up to *maxNumFeatVertex* features (line 3) – we denote such subset by V'. Subsequently, for each vertex $v \in V'$ (line 4) RANKFEATURES determines the associated FTI score and distributes it evenly across its features: more precisely, the algorithm first determines the subset of features associated with v (line 5), then computes the FTI score divided by the number of features associated with v (line 6), and finally updates the score of each feature associated with v (line 7). The algorithm finally concludes by producing the final ranking, by sorting the pairs $(feature, score)$ in *RankFeatures* according to their score (line 8).

We note that the variable *maxFeatsVertex* allows the user to choose a proper trade-off between the amount of information considered from the graph and the execution time that results from the complexity of the cycle at line 7.

4 EXPERIMENTAL SETTING

In this section we provide the experimental setting; more specifically, Section 4.1 introduces the dataset as well as the related data preparation. Section 4.2 provides the competitors, while Section 4.3 introduces the experimental methodology. Finally, Section 4.4 provides the parameters used at run-time with the various competitors.

4.1 Dataset and data preparation

In this work we consider two different kind of dataset:

- Historical audit data covering an interval of time comprised between 2009 and 2011. The data is provided by the Treasury Office of the State of Ceará (SEFAZ-CE - Brazil). From now on, let's name these datasets *TFM-2009*, *TFM-2010* and *TFM-2011*
- Anonymized credit card transactions labeled as fraudulent or genuine. The datasets contains transactions made by credit cards in September 2013 by european cardholders. From here, we'll name this dataset *DS-CreditCard*

About historical audit data, for each year covered we focus on specific subsets of *fraud features* (hereinafter denoted for brevity by *features*), among those available, according to the recommendations provided by expert tax auditors. More precisely, for data covering 2009 and 2010 we focus on a set of 14 features, while for data covering 2011 we focus on 16 features – in this context we note that features frequently change over the years, as fraudsters tend to evolve their strategies. Finally, we report that features and taxpayer identities are anonymized.

Accordingly, we arrange the data in three different tax fraud matrices – Table 2 reports their main characteristics. In the matrices, each row refers to a single taxpayer while each column is associated with a specific fraud feature. As such, if $m_{i,j}$ represents a generic element of the matrices, we have that $m_{i,j} = 1$ if the i-th taxpayer committed the j-th fraud, $m_{i,j} = 0$ otherwise.

Taxpayers who did not present any evidence of fraud were excluded, as they are considered outliers. After having preprocessed the data, the total of each feature for each taxpayer

Dataset	**Year covered**	**# features** (columns)	**# taxpayers** (rows)	**# records Class 1**
TFM-2009	2009	14	11,386	3,632 (31%)
TFM-2010	2010	14	12,424	3,987 (32%)
TFM-2011	2011	16	10,627	3,675 (34%)

Table 2 – Characteristics of the tax fraud matrices considered in the experimental evaluation.

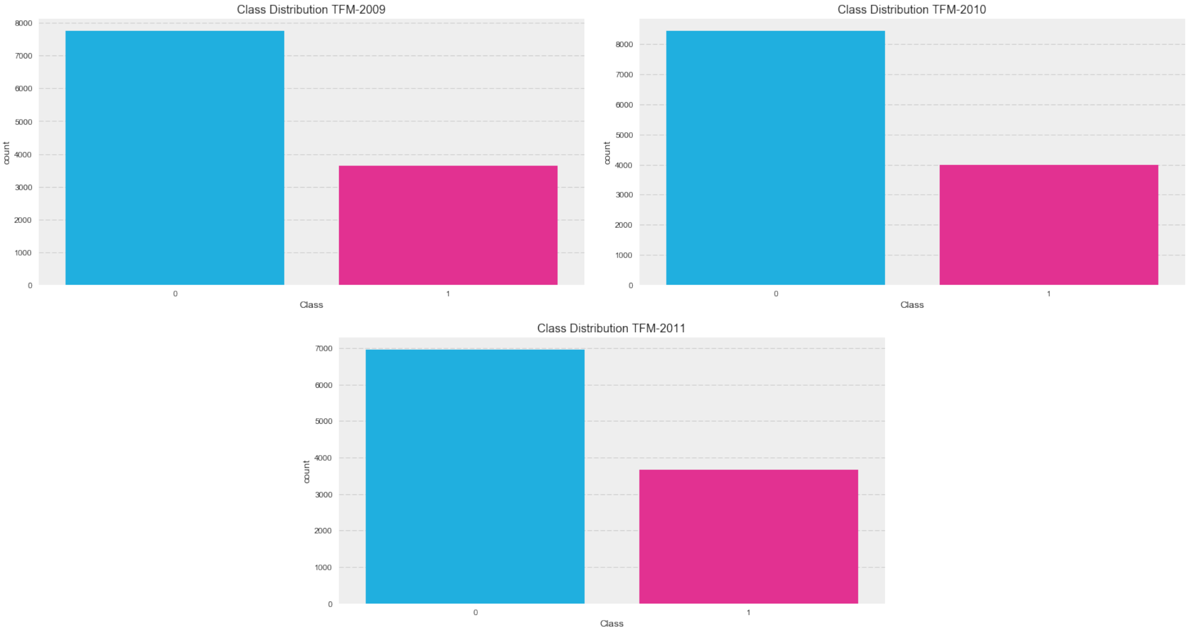

Figure 1 – Class distribution for *TFM-2009*, *TFM-2010*, and *TFM-2011* datasets.

is summed up (aggregated).

4.1.1 Exploratory Data Analysis

In Table 2 we can see the class distribution, represented by the number of records of the class Fraud (column # records Class 1). We can see this distribution represented graphically in Figure 1. From this information we can say that we are working with balanced classes.

Feature correlation is another important analysis: between each other and between a feature and the output class. Assuming as high correlation values ≥ 0.8, we used Pearson correlation (COHEN *et al.*, 2014) for this purpose. This analysis is presented in Figure 2, for *TFM-2009*, which doesn't have high correlation between features. This assumption is true for *TFM-2010* (Figure 3) and *TFM-2011* (Figure 4). This analysis led us to an important finding about the data we are working on, being one of the great reasons for the need to create the new method proposed in this research. We see this translated in the Research Question, in the highlighted section below in bold:

- How do we select the most appropriate subset of binary features, **with low or non-linear correlation between themselves and between them and the targeted class**, such that the tax fraudsters binary classifiers achieve the best performance against state of art competitors?

As a way to test the generalization of our method in other datasets, we used credit

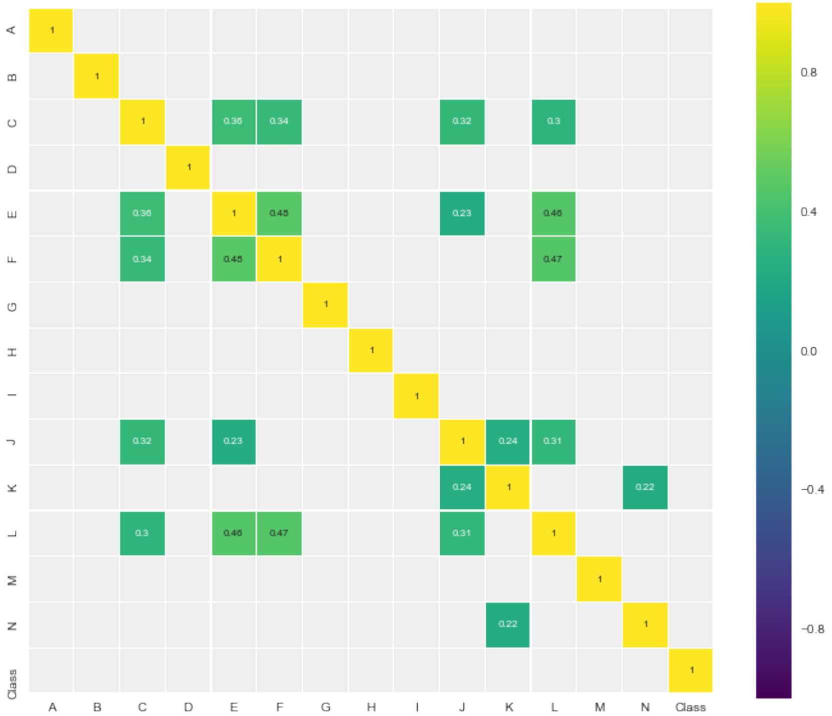

Figure 2 – Feature correlation for *TFM-2009*, using Pearson correlation.

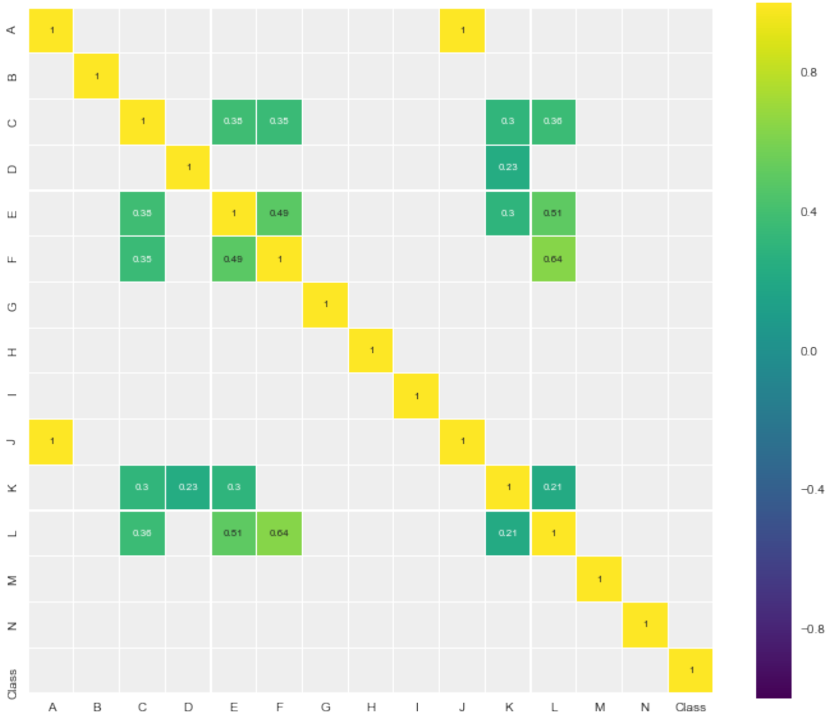

Figure 3 – Feature correlation for *TFM-2010*, using Pearson correlation.

card dataset (*DS-CreditCard*), provided by Kaggle [1]. Although it is not the same domain of our problem, it has some characteristics, such as low correlation between the features, that serve to test our algorithm. This dataset presents transactions that occurred in two days, where we

[1] https://www.kaggle.com/mlg-ulb/creditcardfraud/home (retrieved: 23 June 2018).

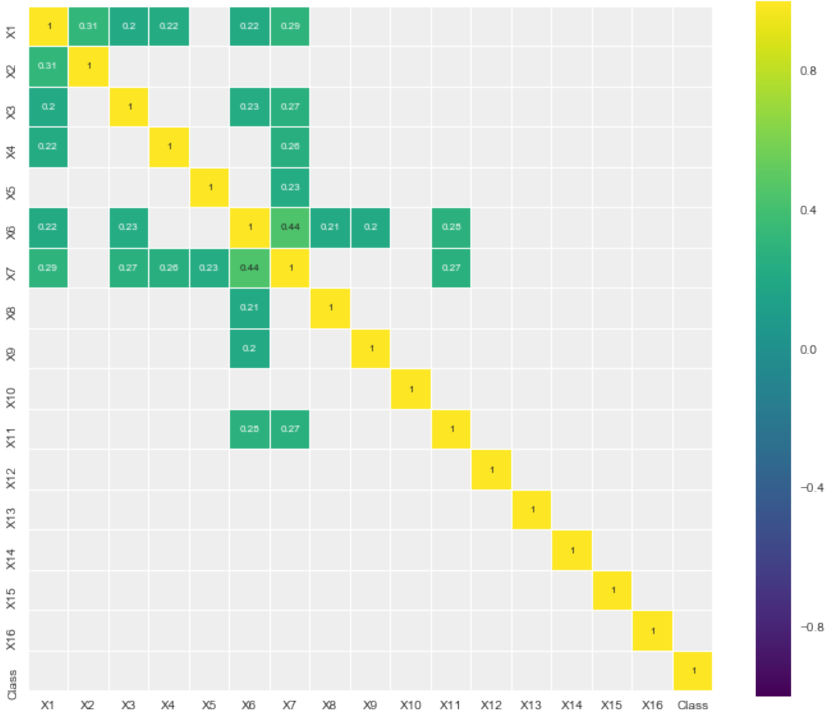

Figure 4 – Feature correlation for *TFM-2011*, using Pearson correlation.

have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

It contains only numerical input variables which are the result of a PCA transformation (POZZOLO *et al.*, 2015). Unfortunately, due to confidentiality issues, they do not provide the original features and more background information about the data. Features V1, V2, ... V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount, this feature can be used for example-dependant cost-sensitive learning. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise. Feature distribution, in *DS-CreditCard*, is presented in Figure 5.

In Figure 6 we can see *DS-CreditCard* doesn't have high correlation between features. These results reinforce the need to adopt a feature selection method to increase the classification for these datasets.

4.2 Competitors

We compare ALICIA with the feature selection methods presented in Section 2.3, i.e., Information Gain (IG), Gain Ratio (GR), Correlation-based Feature Selection (CFS), Feature

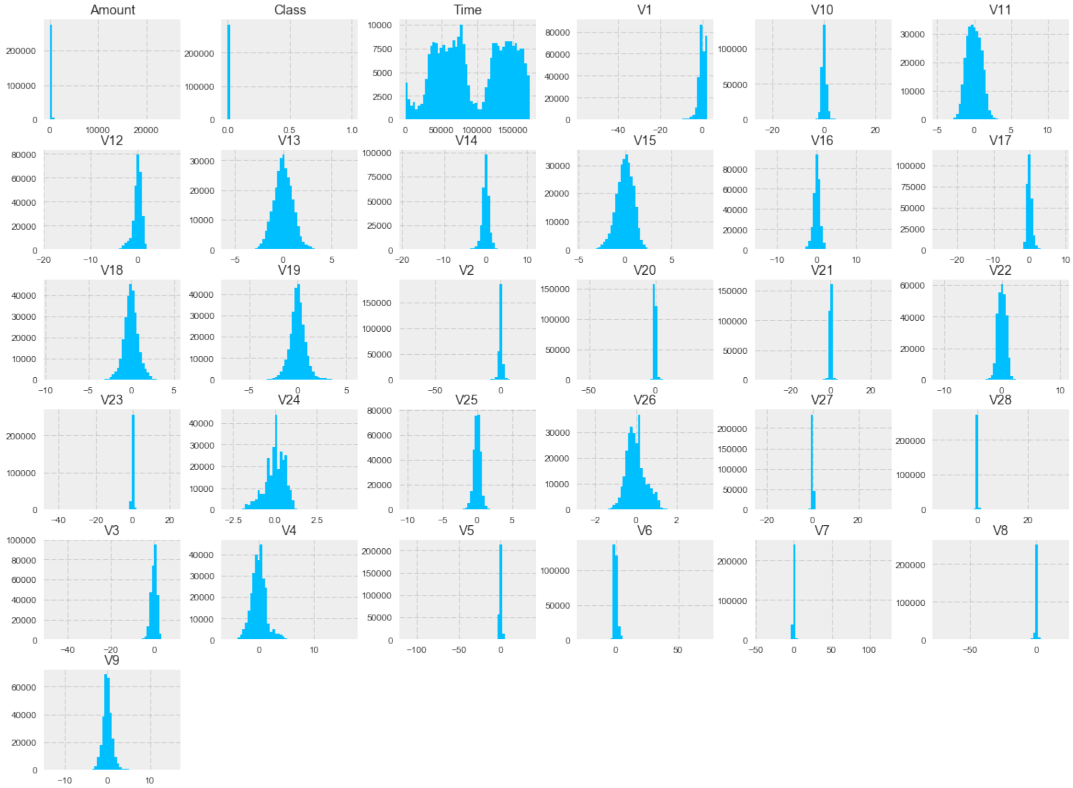

Figure 5 – Features Distribution for *DS-CreditCard*.

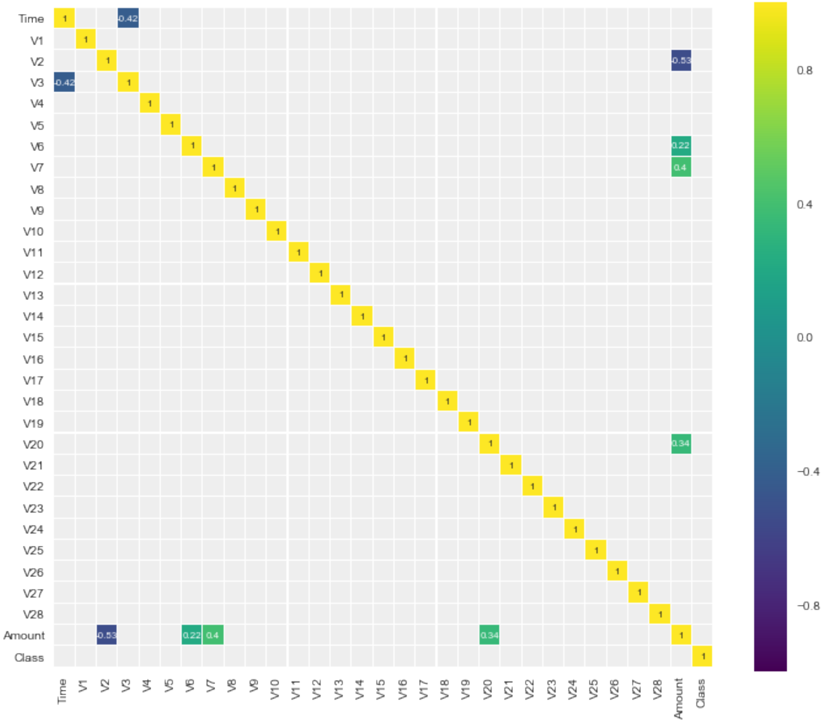

Figure 6 – Feature correlation for *DS-CreditCard*, using Pearson correlation.

Selection via Eigenvector Centrality (ECFS), Relief-F, Fisher, and Gradient boosted feature selection (GBFS).

The research in this paper use the Feature Selection Code Library (FSLib) (ROFFO *et al.*, 2017), for: ECFS, Relief-F and Fisher. In order to analyze IG, GR and CFS we use R package FSelector (ROMANSKI, 2016). We use GBFS code from author, available in (XU, 2018 (Retrieved April 19, 2018)).

For *DS-CreditCard* analysis, we used Python and the best classification algorithm for this task was XGBoost. This algorithm has a native feature selection method: XGBoost feature importance (XGB-FI). Therefore, we also include this method in the comparison. We provide source code on GitHub [2].

4.3 Methodology

To assess the feature selection quality of the various competitors in *TFM-2009*, *TFM-2010* and *TFM-2011*, we employ the SVM-based classifier offered by the SVM package of R (KARATZOGLOU *et al.*, 2004). For *DS-CreditCard* we use XGBoost offered by Python (REFERENCE, 2018 (Retrieved June 23, 2018)).

The classifier is used to determine the recall, precision and F-Measure associated with the results obtained by means of the features selected by the competitors. We also report the use of the 10-fold cross-validation (KOHAVI *et al.*, 1995), as it guarantees less biased estimations of accuracy.

4.4 Run-time parameters

By default *ALICIA* uses the following parameters to generate association rules: $supp = 0.3$, $conf = 0.4$ and $lift = 1.3$ – we report that these values were chosen according to an extensive experimental evaluation (such evaluation is presented in Section 5). For what concerns the other parameters, ALICIA sets by default the n-reachability threshold to 10 and $maxFeaturesVertex$ to 1.

For what concerns the SVM-based classifier, it employs a Radial Basis (Gaussian) kernel, with γ varied in the $[10^{-6}, 10^{-1}]$ range and τ varied in the $[10^{-4}, 10^{-1}]$ range.

For XGBoost classifier, we conducted a cross validation varying parameters in theses

[2] https://github.com/rtales/kaggle-credit-card

ranges: $learning\_rate = [0.05, 0.1], max\_depth = [6, 7, 8], min\_child\_weight = [1, 10], n\_estimators = [400, 500]$. For reproducibility, we used a fixed seed 100.

5 EXPERIMENTAL EVALUATION

In the following we describe the results of the experimental evaluation: Section 5.1 presents an analysis that concerns the generation of association rules from the considered datasets. Section 5.2 presents an analysis on the run-time performance of ALICIA. Finally, Section 5.3 presents a study that compares the quality of the feature selection performed by the various competitors.

5.1 Analysis on the generation of association rules.

The quality of the results returned by association rule learning algorithms depends on the properties of input data and on the choice of proper *support*, *confidence*, and *lift* thresholds. The first parameter on which we focus our attention is *support*, and study how its distribution varies among the association rules generated from the datasets considered. In general, using high support thresholds yields the generation of *relevant* association rules, yet potentially interesting rules involving infrequent features may be discarded. Conversely, low support thresholds allow to overcome the above issue at the expense of generating possibly many *irrelevant* rules. Consequently, studying the support distribution may give insights about a proper trade-off.

In the batch of experiments that follows we vary the support threshold in the $[0.1, 1.0]$ range and report the number of rules generated from the *TFM-2009* dataset. Figure 7, *left plot*, presents the results.

From the Figure we observe that *TFM-2009* exhibits a skewed support distribution, in that the majority of association rules have low support while the remaining ones have high support. From the same plot we also observe a clear distinction in the number of rules generated

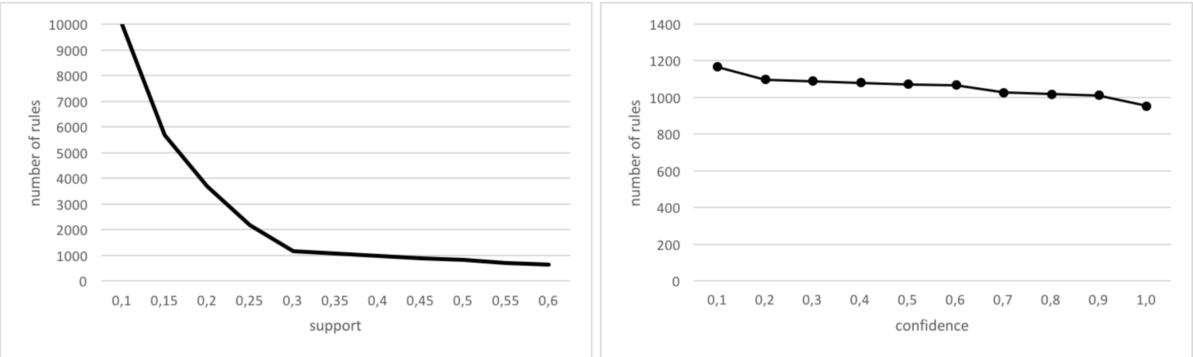

Figure 7 – Analysis on the distributions of support and confidence related to the association rules generated from the *TFM-2009* dataset. *Left plot*: Distribution of support. *Right plot*: distribution of confidence (support is set to 0.3).

when the support threshold is greater or equal than 0.3, indicating that lower values yield the generation of many irrelevant rules – we report that similar results are obtained when considering the *TFM-2010* and *TFM-2011* datasets (results are omitted for brevity). Finally, we report that using a support threshold less or equal than 0.3 allows to generate association rules involving all the features present in the datasets.

In the next batch of experiments we study how varying the confidence affects the generation of association rules. To this end, we fix $support = 0.3$ and variate the confidence in the $[0.1, 1.0]$ range. The dataset considered is, again, *TFM-2009*. Figure 7, *right plot*, presents the results.

From the Figure we observe that variations in *confidence* have minor effects on the number of generated rules. We also report that using a confidence value less or equal than 0.6 allows to produce association rules that involve all the features present in the dataset. Finally, we report that similar results are found when considering the *TFM-2010* and *TFM-2011* datasets (results are omitted for brevity).

In the final batch of experiments we study how varying the lift parameter affects the generation of association rules. To this end, we fix support to 0.3, confidence to 0.6, and variate the lift in the $[1.0, 6.0]$ range. The datasets considered are *TFM-2009*, *TFM-2010*, and *TFM-2011*. Figure 8 presents the results.

From the plots we see that increasing lift has the effect of reducing the number of generated rules, focusing on those that have high correlation between pairs of features – this trend holds for all the three datasets considered. Finally, we report that using a lift value less or equal than 1.3 allows to involve all the features in the generation of association rules.

In light of the experimental findings, in the batches of experiments that follow we always set $support$ to 0.3, $confidence$ to 0.6, and $lift$ to 1.3.

5.2 Analysis on ALICIA's run-time performance

In this study we analyze the run-time performance of ALICIA. We focus on phase III (Section 3.3, Algorithm 4), as the temporal complexity of phase I is dictated by the characteristics of the Apriori algorithm – to this end we refer the reader to (AGRAWAL; SRIKANT, 1994) – while the running time of phase II is negligible. During the third phase there are two parameters that affect the performance of ALICIA: the threshold on the maximum number of features associated with a vertex, $maxFeatsVertex$, and the n-reachability threshold.

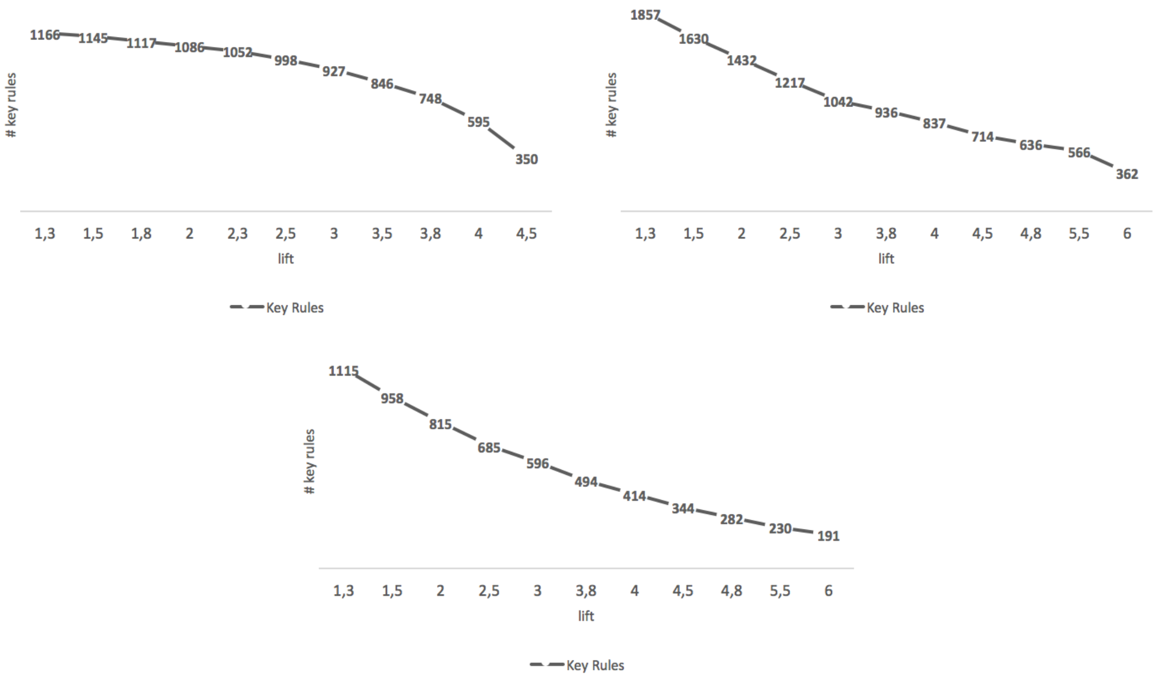

Figure 8 – Distribution of lift, association rules generated from the (a) *TFM-2009*, (b) *TFM-2010*, and (c) *TFM-2011* datasets. $support$ is fixed to 0.3, $confidence$ is fixed to 0.6. Y-axis is associated with the number association rules generated, while X-axis is associated with $lift$ parameter variation.

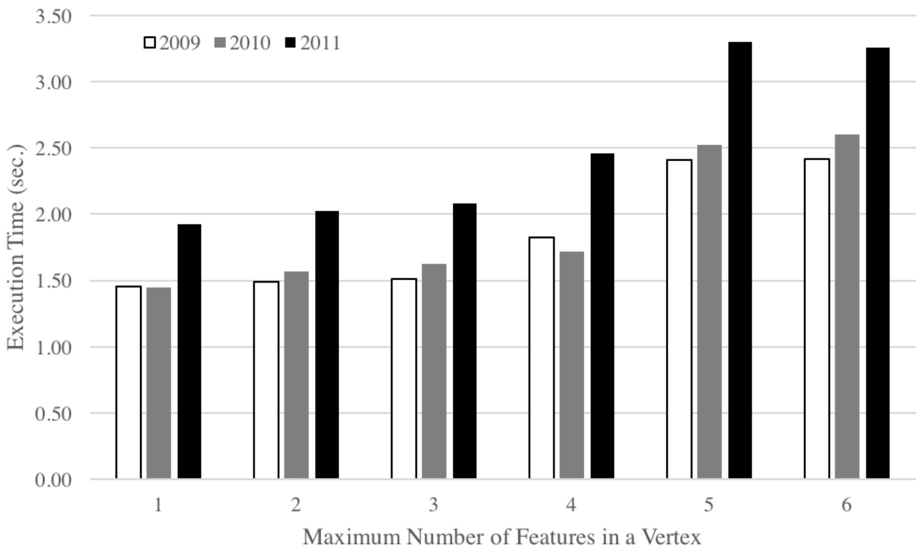

Figure 9 – ALICIA's phase III execution time when varying the *maxFeatsVertex* parameter. The datasets considered are *TFM-2009*, *TFM-2010*, and *TFM-2011*.

In the first batch of experiments we focus on *maxFeatsVertex* and vary it in the [1,14] range when considering the TFM-2009 and TFM-2010 datasets, while we use the [1,16] range when considering the TFM-2011 dataset. All the other parameters are kept fixed to their respective defaults (Section 4.4). Figure 9 reports the results.

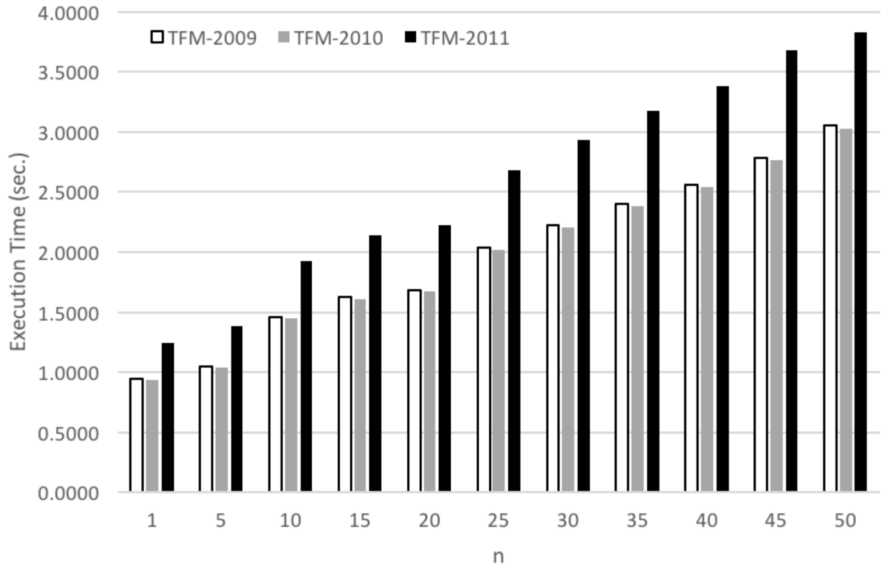

Figure 10 – ALICIA's phase III execution time when varying the *n*-reachability threshold. Y-axis is associated with the execution time (sec.), while X-axis is associated with *n*. The datasets considered are *TFM-2009*, *TFM-2010*, and *TFM-2011*.

From the results, we observe that *maxFeatsVertex* has a noticeable impact on ALICIA's execution time, with increasing running times as *maxFeatsVertex* becomes larger. We also report, however, that the F-measure scores achieved by the SVM-based classifier do not change between $maxFeatsVertex = 1$ and $maxFeatsVertex = 6$, thus indicating that the execution time can be safely reduced by using proper values for *maxFeatsVertex* (according to the characteristics of the dataset considered). We finally note that the maximum value *maxFeatsVertex* could assume during the experiments is 6 as this represents the longest length of the rules produced during phase I.

In the second batch of experiments we focus on the *n*-reachability threshold. To this end, we vary this parameter in the $[1,50]$ range, while keeping the other parameters fixed to their respective defaults (Section 4.4). We use again all the three datasets considered for the purposes of the experimental evaluation. Figure 10 reports the results.

5.3 Analysis on the quality of feature selection

In this Section we assess the quality of the results produced by ALICIA and perform a comparison with its competitors. To this end, we perform two different studies: in the first study we analyze how variations in the *n-reachability* threshold affect the feature selection quality of ALICIA, while in the second study we perform a thorough comparison among the competitors.

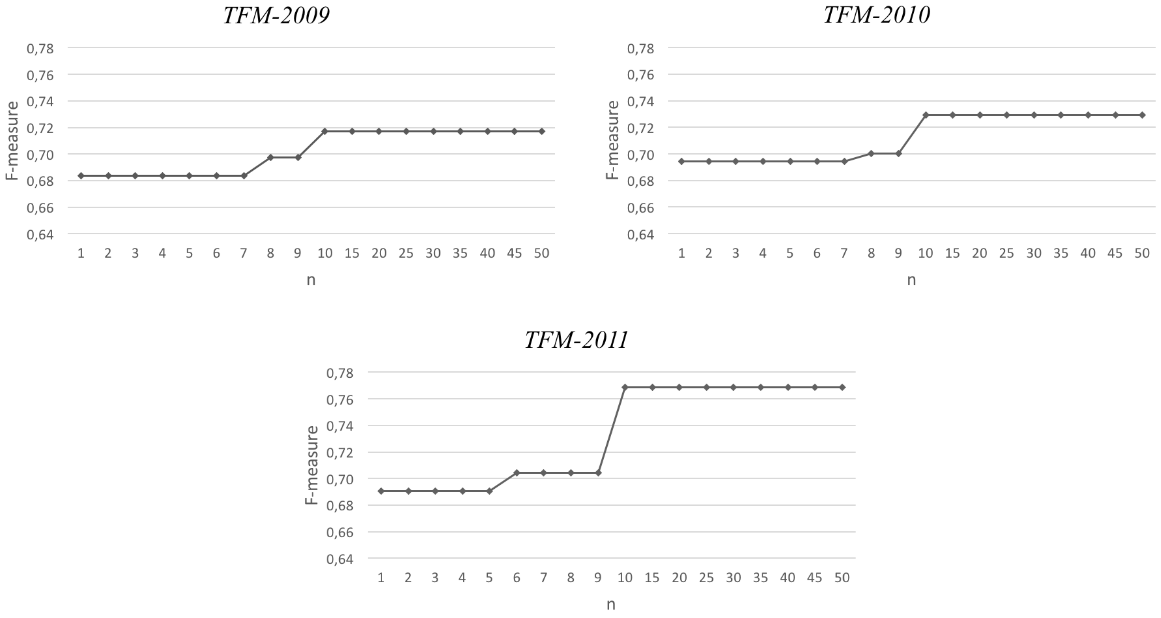

Figure 11 – Analysis on the quality of results when varying the n-reachability within the FTI measure. Y-axis is associated with F-Measure, while X-axis is associated with n. The datasets considered are *TFM-2009*, *TFM-2010*, and *TFM-2011*.

5.3.1 *Varying the n-reachability parameter*

In the first batch of experiments we study how the choice of the n-reachability parameter used with the FTI centrality measure affects the quality of the results returned by ALICIA. To this end we consider all the three datasets and vary n in the $[1,50]$ range; for each value assigned to n we consider the ranking produced by FTI and incrementally include the features, starting from the most relevant ones: the subset of features yielding the best F-Measure determines the F-Measure shown in the plots. For what concerns support, confidence, and lift, we set them to their defaults (Section 4.4). Figure 11 present the results.

From the plots we observe how increasing n improves the F-Measure achieved by ALICIA until $n \leq 10$, while using greater values do not yield further performance gains. This indicates that $n = 10$ represents the best trade-off between the quality that can be achieved and the amount of computations required by FTI. As such, in the batches of experiments that follow we set $n = 10$.

5.3.2 *Competitors comparison*

In the batch of experiments that follows we compare the quality of the results between ALICIA and its competitors. To this end, we consider all the datasets previously introduced, i.e., *TFM-2009*, *TFM-2010*, and *TFM-2011*. We also set $n = 10$ and support, confidence, and lift to their respective defaults (see also Section 4.4). The quality of the results is assessed by means of the F-Measure achieved by the SVM classifier.

5.3.3 Competitors comparison - centrality measure algorithms

In this section, we analyze the results obtained by creating a ranking of the importance of features from the most traditional centrality algorithms compared to our new centrality strategy (FTI) proposed in this thesis. We compare degree centrality, closeness centrality and betweenness centrality against FTI.

From the analysis of the figure 12, it is clear that the ranking of features formed by the FTI strategy obtained the best results when compared to the other centrality strategies.

We can see that betweenness centrality presents the worst results for all datasets. We can also observe that the results have a similar behavior since the values of F-Measure increase or decrease in a similar way for all the metrics according to the number of features used for the calculation. For example, the values decreases for all methods in *TFM-2009* when we reduce from five to four features. We can also observe that, for *TFM-2010*, while all other measures of

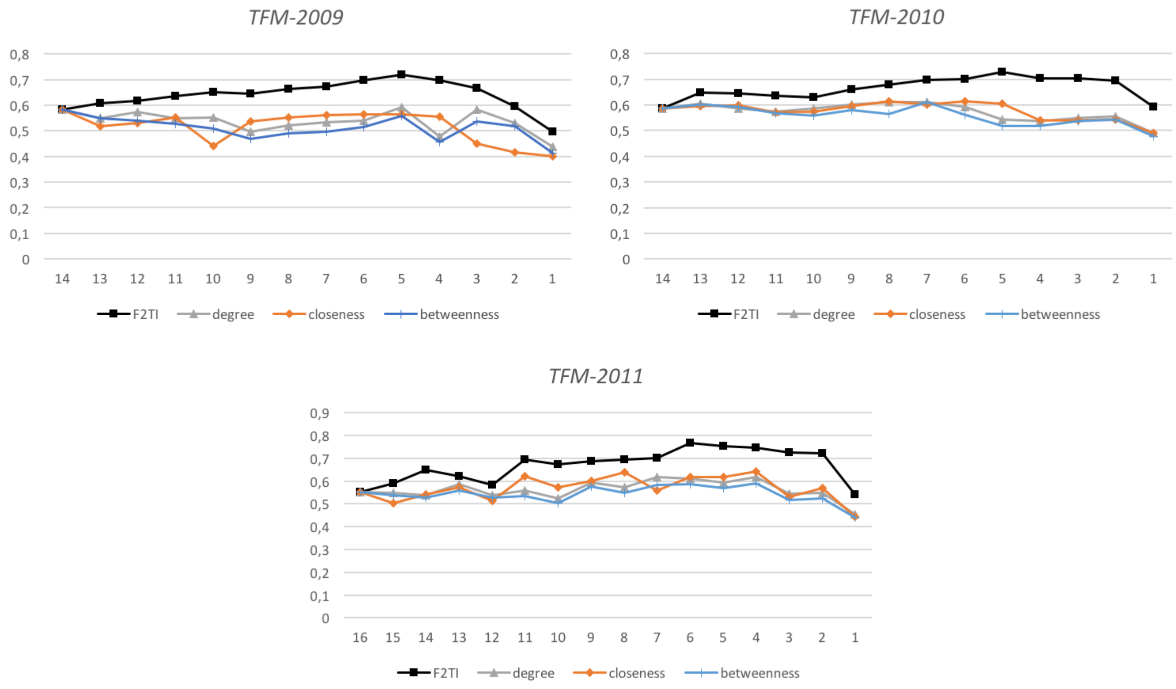

Figure 12 – Competitor comparison: centrality measure algorithms used to rank features. Y-axis is associated with F-Measure, while X-axis is associated with the number of features used to calculate F-measure. The datasets considered are *TFM-2009*, *TFM-2010*, and *TFM-2011*.

centrality present a reduction of the F-Measure from seven features, our method continues to improve until the reduction to five features (when the best result is obtained).

5.3.4 Competitors comparison - feature selection algorithms

The features considered by the various competitors are chosen according to the ranking produced by *FTI* – more precisely, features are incrementally included starting from the most relevant ones. Tables 3, 4, and 5 report the results.

From the tables 3, 4 and 5, we observe that ALICIA almost always outperforms its competitors; more precisely, ALICIA achieves the best results with *TFM-2009* when the number of considered features is equal to 5 (the resulting F-Measure is equal to 72%). Similarly, ALICIA achieves the best results with *TFM-2010* and *TFM-2011* when the number of considered features is, respectively, equal to 5 and 6, with the resulting F-Measure scores equal to 73% and 77%.

It should be also noted, from the results of Tables 3, 4 and 5, that **GR** and **CFS** methods do not achieve the best results in any of the datasets, for any number of features. **IG** method does not present any case of better result for the *TFM-2009* and *TFM-2010* datasets and presents only one better result for *TFM-2011* - when thirteen features are used - even so, this value is 10% lower than the best result obtained with our method.

When we analyze the results obtained by the **ECFS**, **RELIEF-F** and **FISHER** methods, we can see that they only achieve better results in one of the three datasets used in the experiments - *TFM-2009*, *TFM-2010* and *TFM-2009* respectively. None of them achieve better results when applied to dataset *TFM-2011*.

top feats	**IG**	**GR**	**CFS**	**ECFS**	**RELIEF-F**	**FISHER**	**GBFS**	**ALICIA**
14	0.5838	0.5838	0.5838	0.5838	0.5838	0.5838	0.5838	0.5838
13	0.5965	0.5519	0.5937	0.5419	0.5762	0.5647	0.5826	**0.6068**
12	0.6113	0.5766	0.5878	**0.6278**	0.5015	0.5314	0.5581	0.6171
11	0.6027	0.5885	0.5582	**0.6422**	0.5552	0.5895	0.6418	0.6363
10	0.5885	0.4737	0.5497	0.4372	0.5381	0.6243	**0.6546**	0.6508
9	0.5458	0.5769	0.5026	0.4401	0.5663	**0.6512**	0.6398	0.6449
8	0.5671	0.5924	0.5364	0.4698	0.5332	**0.6785**	0.6581	0.6632
7	0.5671	0.5956	0.6088	0.5942	0.4707	0.6213	0.6006	**0.6723**
6	0.5856	0.6130	0.5322	0.5176	0.5645	0.5819	0.5675	**0.6956**
5	0.6374	0.6001	0.5508	0.5044	0.5476	0.5512	0.6247	**0.7172**
4	0.5261	0.6022	0.4918	0.4873	0.4879	0.5416	0.5051	**0.6975**
3	0.6251	0.4904	0.5875	0.4691	0.4261	0.5318	0.6001	**0.6661**
2	0.5844	0.4114	0.5456	0.4182	0.4368	0.5115	0.5786	**0.5938**
1	0.4717	0.4047	0.4426	0.4102	0.4061	0.4811	0.4623	**0.4966**

Table 3 – Analysis on the quality of feature selection, **TFM-2009** dataset. The column **top feats** illustrates the number of *top* features selected from the ranking produced by the various feature selection algorithms. Each figure within the other columns represents the F-measure achieved by the SVM-based classifier when using the ranking produced by the associated feature selection method. Winners are highlighted in **bold**.

top feats	IG	GR	CFS	ECFS	RELIEF-F	Fisher	GBFS	ALICIA
14	0.5868	0.5868	0.5868	0.5868	0.5868	0.5868	0.5868	0.5868
13	0.6468	0.6359	0.6189	0.4245	0.5584	0.5803	**0.6515**	0.6485
12	0.6355	0.6429	0.6220	0.5916	0.5848	0.6092	**0.6496**	0.6434
11	0.6277	0.6236	0.6273	0.5515	**0.6512**	0.5483	0.6292	0.6342
10	0.6253	0.6164	0.6148	0.4749	**0.6348**	0.5715	0.6108	0.6297
9	0.6514	0.6513	0.6427	0.4563	0.4302	0.6424	0.6371	**0.6599**
8	0.6548	0.6607	0.6406	0.5202	0.4743	0.6281	0.6316	**0.6784**
7	0.6592	0.6512	0.6414	0.5422	0.4621	0.6103	0.6295	**0.6961**
6	0.6362	0.6560	0.6396	0.5214	0.5139	0.6202	0.6201	**0.7006**
5	0.5828	0.6499	0.6545	0.4573	0.6018	0.6099	0.5915	**0.7291**
4	0.5736	0.5919	0.5816	0.4084	0.4456	0.5893	0.5730	**0.7044**
3	0.5963	0.5906	0.5803	0.4150	0.4650	0.5708	0.5879	**0.7029**
2	0.6081	0.5912	0.5831	0.4288	0.4078	0.5691	0.5718	**0.6956**
1	0.5301	0.5228	0.5159	0.4824	0.5049	0.5681	0.5477	**0.5915**

Table 4 – Analysis on the quality of feature selection, **TFM-2010** dataset.

top feats	IG	GR	CFS	ECFS	RELIEF-F	Fisher	GBFS	ALICIA
16	0.5534	0.5534	0.5534	0.5534	0.5534	0.5534	0.5534	0.5534
15	0.5893	0.5579	0.5452	0.5854	0.5493	0.5667	**0.5957**	0.5905
14	0.5765	0.5982	0.528	0.6105	0.5431	0.5265	**0.6614**	0.6501
13	**0.6481**	0.6086	0.5961	0.6201	0.6210	0.5542	0.6221	0.6199
12	0.5945	0.5572	0.5519	0.5389	0.5637	0.5422	**0.6016**	0.5824
11	0.6084	0.6683	0.5486	0.5482	0.6000	0.5406	0.6227	**0.6937**
10	0.5805	0.6315	0.5258	0.6088	0.5457	0.5217	0.5929	**0.6724**
9	0.6523	0.6591	0.5915	0.5783	0.6099	0.5328	0.6212	**0.6877**
8	0.6264	0.6832	0.5702	0.5345	0.5231	0.5638	0.5913	**0.6948**
7	0.6671	0.6004	0.6025	0.5227	0.5239	0.5656	0.6418	**0.7023**
6	0.6608	0.6706	0.6111	0.6339	0.5428	0.5115	0.6463	**0.7688**
5	0.6433	0.6634	0.6119	0.5436	0.6273	0.5583	0.6235	**0.7534**
4	0.6709	0.6976	0.5937	0.6284	0.5819	0.5124	0.6451	**0.7468**
3	0.6002	0.5752	0.5491	0.6015	0.5986	0.5244	0.5833	**0.7260**
2	0.6046	0.6089	0.5588	0.6068	0.5250	0.5576	0.5798	**0.7211**
1	0.4987	0.4915	0.4611	0.5157	0.6091	0.5144	0.5055	**0.5421**

Table 5 – Analysis on the quality of feature selection, **TFM-2011** dataset.

In this way, we can see that only **ALICIA** achieves better results in all datasets. In this way, we can observe that among the filtering methods, only **ALICIA** achieves better results in all datasets. In addition, only **ALICIA** maintains the best results in all datasets when the number of features is reduced by less than half. Considering that the best results were obtained when we have less than half of the features used, our method is the best option among the others filtering methods.

We also compared our strategy against the **GBFS** algorithm, because although it is a wrapper method it is considered the state of the art in feature selection. In order to obtain the best result with GBFS, we tested it with several combinations of parameters: $learningrate = [0.01, 0.05, 0.1]$, $depth = [4, 5, 6]$, $ntrees = [100, 200, 300, 400]$.

Although **GBFS** achieved some better result in all datasets, in none of them did it get the best overall result and in none of them was it better when there was a reduction of at least half of the features.

The best result for GBFS was obtained with the following parameters: $learningrate = 0.05$, $depth = 5$, $ntrees = 200$.

top feats	IG	GR	XGB-FI	CFS	ECFS	RELIEF-F	Fisher	GBFS	ALICIA
30	0.8696	0.8696	0.8696	0.8696	0.8696	0.8696	0.8696	0.8696	0.8696
29	0.8663	0.8663	0.8663	0.8649	0.8696	0.8710	0.8663	0.8649	**0.8743**
28	0.8710	0.8710	0.8865	0.8268	0.8649	0.8710	0.8710	0.8757	**0.8865**
27	0.8466	0.8617	0.8770	0.8161	0.8649	0.8710	0.8617	0.8757	**0.8865**
26	0.8571	0.8617	0.8663	0.8315	0.8541	0.8710	0.8602	0.8804	**0.8865**
25	0.8571	0.8617	0.8663	0.8202	0.8710	0.8710	0.8602	0.8663	**0.8913**
24	0.8511	0.8617	0.8723	0.8268	0.8462	**0.8757**	0.8663	0.8511	0.8427
23	0.8723	0.8466	0.8663	0.8268	0.8444	**0.8757**	0.8602	0.8710	0.8315
22	0.8723	**0.8757**	0.8617	0.8380	0.8444	0.8617	**0.8757**	0.8696	0.8380
21	0.8723	0.8710	0.8723	0.8444	0.8444	0.8617	**0.8852**	0.8710	0.8492
20	0.8632	0.8617	0.8571	0.8427	0.8444	0.8511	**0.8696**	0.8587	0.8380
19	0.8663	0.8663	0.8360	0.8398	0.8462	0.8466	**0.8830**	0.8757	0.8398
18	0.8617	0.8663	0.8360	0.8242	0.8444	0.8617	**0.8770**	**0.8770**	0.8539
17	0.8511	0.8602	0.8404	0.8398	0.8539	0.8387	**0.8770**	0.8617	0.8603
16	0.8449	**0.8649**	0.8387	0.8370	0.8427	0.8324	0.8617	**0.8649**	0.8619
15	0.8387	**0.8817**	0.8298	0.8495	0.8475	0.8541	0.8541	0.8602	0.8539
14	0.8280	**0.8757**	0.8404	0.8495	0.8475	0.8541	0.8602	0.8663	0.8588
13	0.8404	**0.8791**	0.8360	0.8649	0.8475	0.8541	0.8710	0.8723	0.8603
12	0.8556	0.8602	0.8449	0.8352	0.8475	0.8541	0.8587	**0.8663**	0.8556
11	0.8602	0.8511	0.8449	0.8333	0.8588	0.8602	0.8587	0.8602	**0.8619**
10	0.8415	0.8415	0.8495	0.8380	0.8556	0.8696	0.8398	0.8495	0.8556
9	0.8478	0.8462	0.8556	0.8333	0.8352	0.8541	0.8404	0.8495	0.8462
8	0.8432	0.8587	0.8602	0.8023	0.8045	0.8743	0.8478	0.8525	0.8398
7	0.8197	0.8148	0.8634	0.8023	0.8043	0.8696	0.8432	0.8619	0.8508
6	0.8202	0.7568	0.8541	0.7126	0.7778	0.8478	0.8370	0.8360	0.8045
5	0.8085	0.7708	0.8462	0.6857	0.7869	0.8387	0.8387	0.8085	0.8045
4	0.8085	0.7813	0.8132	0.6826	0.5867	0.8235	0.7979	0.8085	0.8156
3	0.7937	0.7857	0.8415	0.0594	0.5503	0.8415	0.7514	0.7937	0.8068
2	0.7486	0.7449	0.7351	0.0000	0.2414	0.7351	0.6429	0.7486	0.7174
1	0.6882	0.6882	0.6023	0.0000	0.0000	0.6023	0.6000	0.6882	0.6000

Table 6 – Analysis on the quality of feature selection, *DS-CreditCard* dataset. The column **top feats** illustrates the number of *top* features selected from the ranking produced by the various feature selection algorithms. Each figure within the other columns represents the F-measure achieved by the XGB classifier when using the ranking produced by the associated feature selection method. Winners are highlighted in **bold**.

In order to analyze how much our method generalizes to other dataset types, we apply our method and compare it with the competitors in a unbalanced dataset: *DS-CreditCard*. In this analysis, we have included an over-comparison method XGBoost Feature Importance (XGB-FI). This was necessary because here we apply Python XGBoost and this method is native to this classifier.

The results obtained with each method, using the *DS-CreditCard*, is presented in Table 6, where we observe that ALICIA achieves the best classification result when the number of considered features is equal to 25 (the resulting F-Measure is equal to 89.13%). This result outperforms all the competitors.

5.3.5 Related work comparison - feature selection algorithms

In this section we used the methods described in section 2.1 to perform classification in our datasets *TFM-2009*, *TFM-2010* and *TFM-2011*, in order to compare with the results using

FS methods	Classification algorithm	Dataset	# features	# reduced features	Metric	Classification result
FAWCETT; PROVOST, 1997	Neural network	*TFM-2009*	14	8	F-measure	59%
		TFM-2010	14	8		62%
		TFM-2011	16	11		60%
YANG; HWANG, 2006	Markov blanket filter	*TFM-2009*	14	11		58%
		TFM-2010	14	10		60%
		TFM-2011	16	12		59%
KOTSIANTIS et al., 2006	C4.5	*TFM-2009*	14	8		58%
		TFM-2010	14	9		60%
		TFM-2011	16	11		59%
BELHADJI et al., 2000	Probit model	*TFM-2009*	14	9		49%
		TFM-2010	14	9		52%
		TFM-2011	16	13		51%
ALICIA	SVM	*TFM-2009*	14	9		**72**%
		TFM-2010	14	9		**73**%
		TFM-2011	16	13		**77**%

Table 7 – Analysis on the quality of Feature selection methods, presented in Related work section, applied to our dataset.

our method. The results are presented in Table 7.

From the results, it should be noted that in all methods, the best result could only be achieved with more than half of the existing features and all of them obtained results significantly lower than those obtained with our method, as we observed in the previous section.

In conclusion, we can assume that our centrality measure FTI is more successful in mapping the relationship between features and our feature selection method **ALICIA** surpass all the compared competitors. In this way, we recommend using our method especially in the following situations: (i) when the dataset has binary independent variables or that can be transformed into binary, (ii) when the correlation between the independent variables is not high, and (iii) when the dependent variable is binary (binary class).

As explained, the results presented were obtained with an SVM classifier. However, other classifiers were also tested: Random Forest, Decision Tree and XGBoost. Of these, XGBoost was the one that most approached the results obtained with the SVM. The following is an analysis to check if the results presented by SVM and XGBoost are statistically significant or not.

5.3.6 Statistical Comparison of Classifiers

In this section we performed a Statistical evaluation of our results comparing SVM and XGBoost classifiers on each dataset used in this thesis. We have experimental results from two different algorithms (SVM and XGBoost) for each dataset. Each algorithm has been trialed multiple times on the test dataset and a skill score has been collected. We are left with two populations of skill scores for each dataset.

We know that the data represents an error score on a test dataset and that minimizing

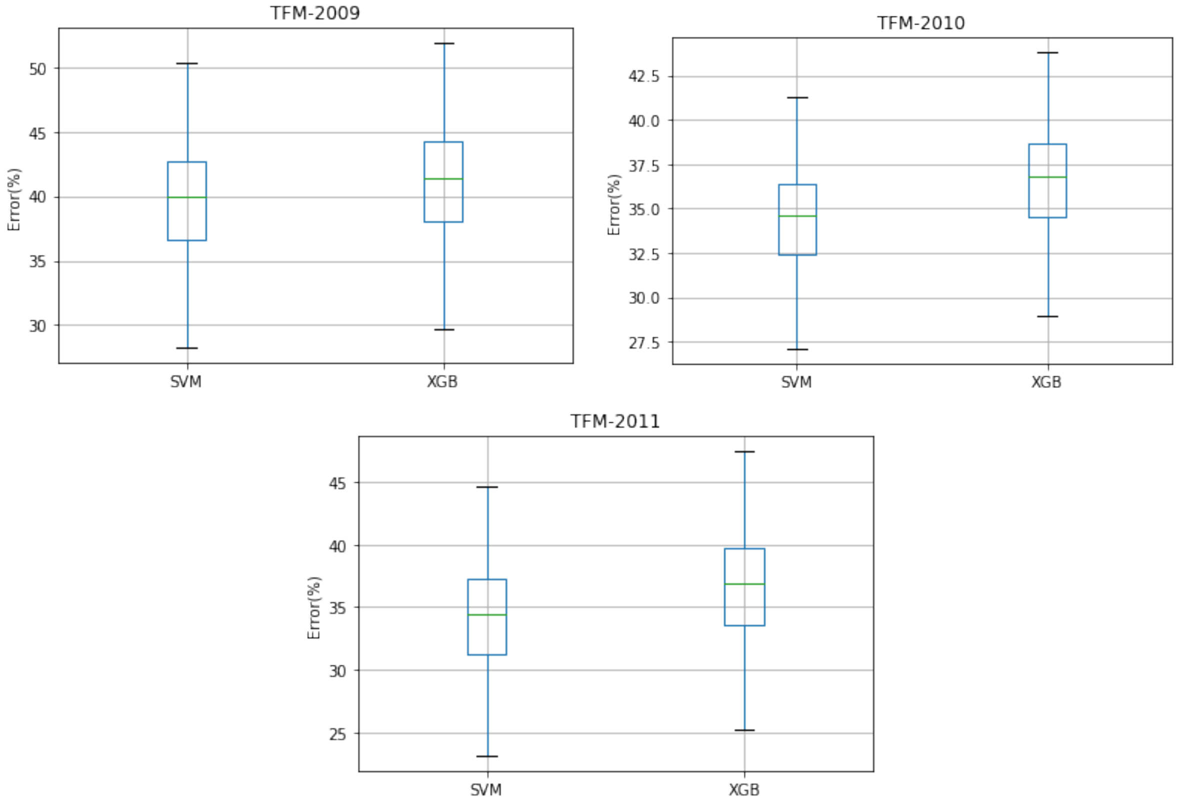

Figure 13 – Classification error. Y-axis is associated with classification error, while X-axis is associated with classification algorithm. The datasets considered are *TFM-2009*, *TFM-2010*, and *TFM-2011*.

the score is the goal.

We can see in table 8 that on average SVM was better than XGBoost in **TFM-2009**, **TFM-2010** and **TFM-2011**. We can also see the same story in the median (50th percentile). Looking at the standard deviations, we can also see that it appears both distributions have a similar (identical) spread (Figure 13).

	2009		**2010**		**2011**	
	SVM	**XGBoost**	**SVM**	**XGBoost**	**SVM**	**XGBoost**
COUNT	100.0	100.0	100.0	100.0	100.0	100.0
MEAN	35.30	36.76	34.58	36.76	34.46	36.91
STD	2.64	2.66	2.82	2.94	4.27	4.41
MIN	28.28	29.67	27.09	28.94	23.12	25.19
25%	33.30	34.73	32.44	34.52	31.22	33.56
50%	35.31	36.77	34.59	36.77	34.47	36.92
75%	37.01	38.49	36.41	38.66	37.22	39.77
MAX	41.62	43.13	41.32	43.79	44.66	47.45

Table 8 – Descriptive statistics that summarize the central tendency, dispersion and shape of all datasets' distribution, excluding NaN values.

5.3.6.1 Normaly test

The determination of distribution type is necessary to determine the critical value and test to be chosen to validate any hypothesis. We can use a statistical test to confirm that the results drawn from distributions are Gaussian (also called the normal distribution).

(H0) null hypothesis of the test:

The null hypothesis of the test ($H0$), or the default expectation, is that the statistic describes a normal distribution. We fail to reject this hypothesis if the p-value is greater than 0.05. We reject this hypothesis if the p-value ≤ 0.05. In this case, we would believe the distribution is not normal with 95% confidence.

The benefit of using p-value is that it calculates a probability estimate, we can test at any desired level of significance by comparing this probability directly with the significance level. Our test tests the null hypothesis that a sample comes from a normal distribution. It is based on D'Agostino and Pearson's (D'AGOSTINO, 1971; D'AGOSTINO; PEARSON, 1973) test that combines skew and kurtosis to produce an omnibus test of normality. We can see in Table 9, that fail to reject $H0$ and that distribution for error in **TFM-2009**, **TFM-2010** and **TFM-2011** is normal for both SVM and XGBoost. An histogram with these distribution, can be seen in Figure 14.

In other words, sets of results are Gaussian and have the same variance; this means we can use the Student t-test to see if the difference between the means of the two distributions is statistically significant or not.

5.3.6.2 Compare Means for Gaussian Results

A t-test is used to compare the mean of two given samples. It assumes a normal distribution of the sample.

(H0) null hypothesis of the test:

The null hypothesis of the test ($H0$) or the default expectation is that both samples

	SVM	**XGBoost**
	p-value	**p-value**
TFM-2009	0.9502	0.9502
TFM-2010	0.9502	0.9502
TFM-2011	0.9502	0.9502

Table 9 – Distribution analysis for classification error, using P-value and confidence.

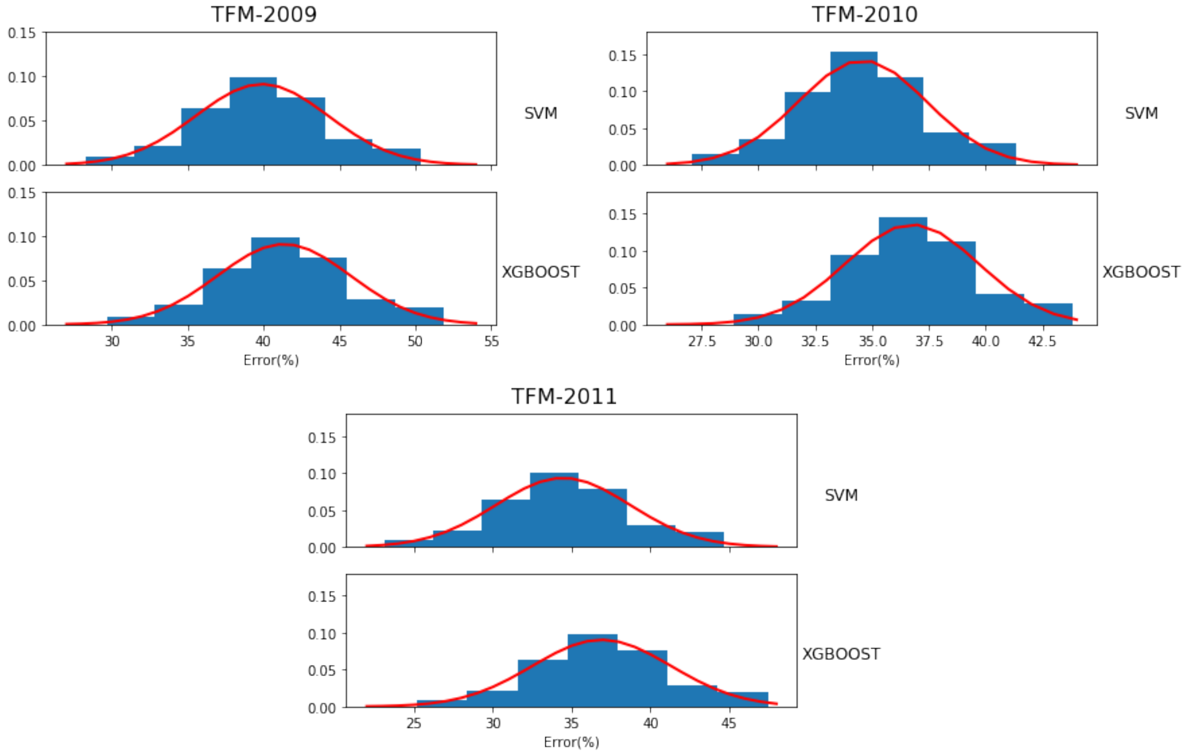

Figure 14 – Error distribution histogram. Y-axis is associated with distribution bins, while X-axis is associated with classification error. The datasets considered are *TFM-2009*, *TFM-2010*, and *TFM-2011*.

	p-value	**T-statistic**
TFM-2009	0.0202631351804	-2.34031207733
TFM-2010	2.5650898687e-07	-5.3376797832
TFM-2011	9.31733001998e-05	-3.98947926541

Table 10 – Difference between the means of SVM and XGBoost, using P-value and T-test.

were drawn from the same population. If we fail to reject this hypothesis, it means that there is no significant difference between the means.

If we get a p-value of ≤ 0.05, it means that we can reject the null hypothesis and that the means are significantly different with a 95% confidence.

Running the example prints the statistic and the p-value. We can see that the p-value is much lower than 0.05. In fact, it is so small that we have a near certainty that the difference between the means is statistically significant. We can reject $H0$, since samples are likely drawn from different distributions.

The closer the distributions are, the larger the sample that is required to tell them apart. We can demonstrate this by calculating the statistical test on different sized sub-samples of each set of results and plotting the p-values against the sample size.

We would expect the p-value to get smaller with the increase sample size. In Figure

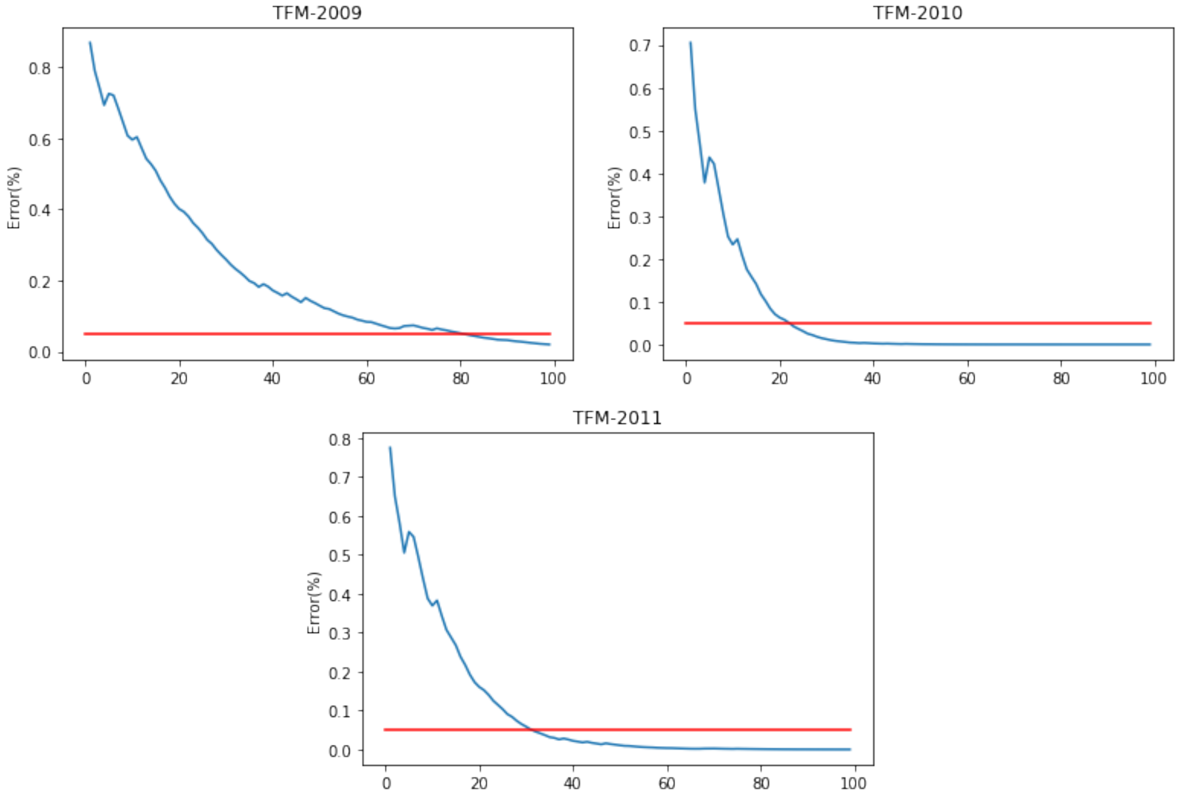

Figure 15 – At what point the sample size is large enough to indicate these two populations are significantly different. Y-axis is associated with classification error, while X-axis is associated with sample size. The datasets considered are *TFM-2009*, *TFM-2010*, and *TFM-2011*.

15 we can also draw a line at the 95% level (0.05) and show at what point the sample size is large enough to indicate these two populations are significantly different. In this way, we suggest the use of an SVM-based classifier.

5.3.6.3 Precision-Recall and ROC Curves Analysis

Receiver Operator Characteristic curves (ROC) and Precision-Recall curves (PR) are commonly used to present results for binary decision problems in machine Learning (PROVOST *et al.*, 1998). For *TFM-2009*, *TFM-2010*, and *TFM-2011* where the class distribution is close to be uniform, ROC curves have many desirable properties. However, when dealing with a highly skewed dataset as *DS-CreditCard*, PR curves give a more accurate picture of an algorithm's performance (SAITO; REHMSMEIER, 2015). In other words, The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets.

An important difference between ROC space and PR space is the visual representation of the curves. The goal in ROC space is to be in the upper-left-hand corner and goal in PR space is to be in the upper-right-hand corner.

The ROC curve shows the tradeoff between specificity and sensitivity (FAWCETT, 2006). It is model-wide because it shows pairs of specificity and sensitivity values calculated at all possible threshold scores. In ROC curves, classifiers with random performance show a straight diagonal line from (0, 0) to (1, 1) (FAWCETT, 2006), and this line can be defined as the baseline of ROC. A ROC curve provides a single performance measure called the Area under the ROC curve (AUC) score. AUC is 0.5 for random and 1.0 for perfect classifiers (HANLEY; MCNEIL, 1982).

In binary classification, data is divided into two different classes, positives (P) and negatives (N). The binary classifier then classifies all data instances as either positive or negative. This classification produces four types of outcome—two types of correct (or true) classification, True Positives (TP) and True Negatives (TN), and two types of incorrect (or false) classification, False Positives (FP) and False Negatives (FN). A 2x2 table formulated with these four outcomes is called a confusion matrix. All the basic evaluation measures of binary classification are derived from the confusion matrix.

The PR curve shows the relationship between precision and sensitivity, and its baseline moves with class distribution. The PR shows precision values for corresponding sensitivity (recall) values. Similar to the ROC curve, the PR curve provides a model-wide evaluation.

While the baseline is fixed with ROC, the baseline of PR is determined by the ratio of positives (P) and negatives (N) as $y = P/(P+N)$. Because of this moving baseline, AUC (PR) also changes with the P:N ratio. For instance, the AUC (PR) of random classifiers is 0.5 only for balanced class distributions, whereas it is $P/(P+N)$ for the general case, including balanced and imbalanced distributions. In fact, AUC (PR) is identical to the y-position of the PR baseline.

The visual interpretability of ROC plots in the context of imbalanced datasets can be deceptive with respect to conclusions about the reliability of classification performance, owing to an intuitive but wrong interpretation of specificity. PR plots, on the other hand, can provide the viewer with an accurate prediction of future classification performance due to the fact that they evaluate the fraction of true positives among positive predictions. Let us consider Credit Card fraud detection, using *DS-CreditCard* dataset. When we one looks at the ROC curve in Figure 19 it appears to be fairly close to optimal ($AUC = 0.95$), and the PR curve shows that there is still room for improvement ($AP = 0.83$).

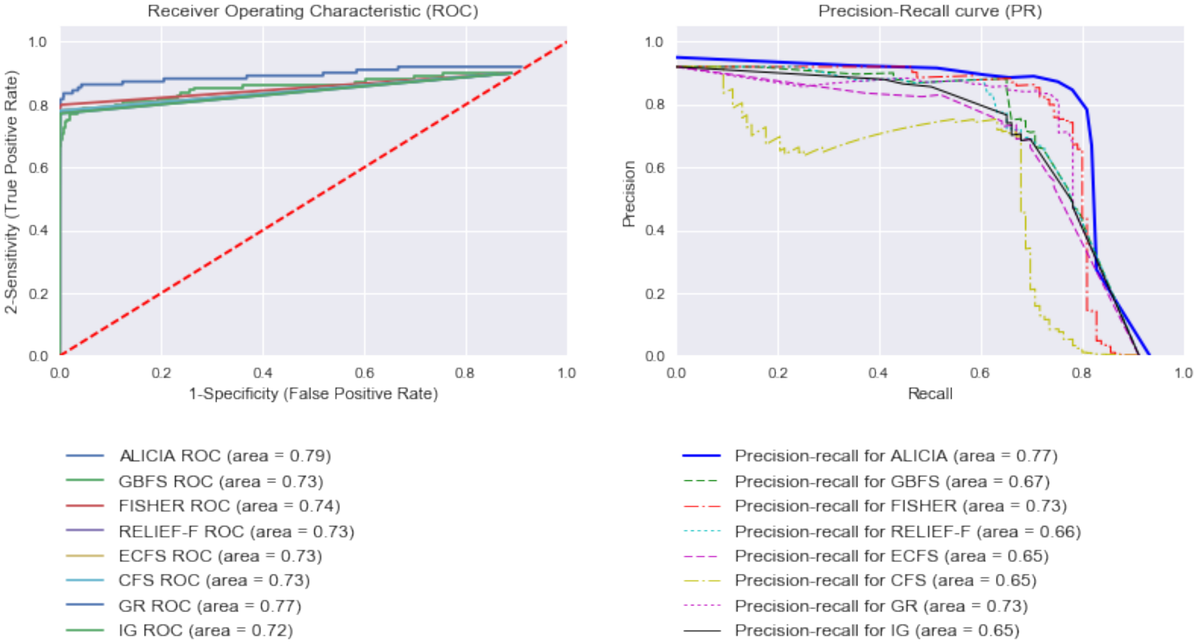

Figure 16 – The same curve shown in both ROC and PR space, for *TFM-2009* dataset.

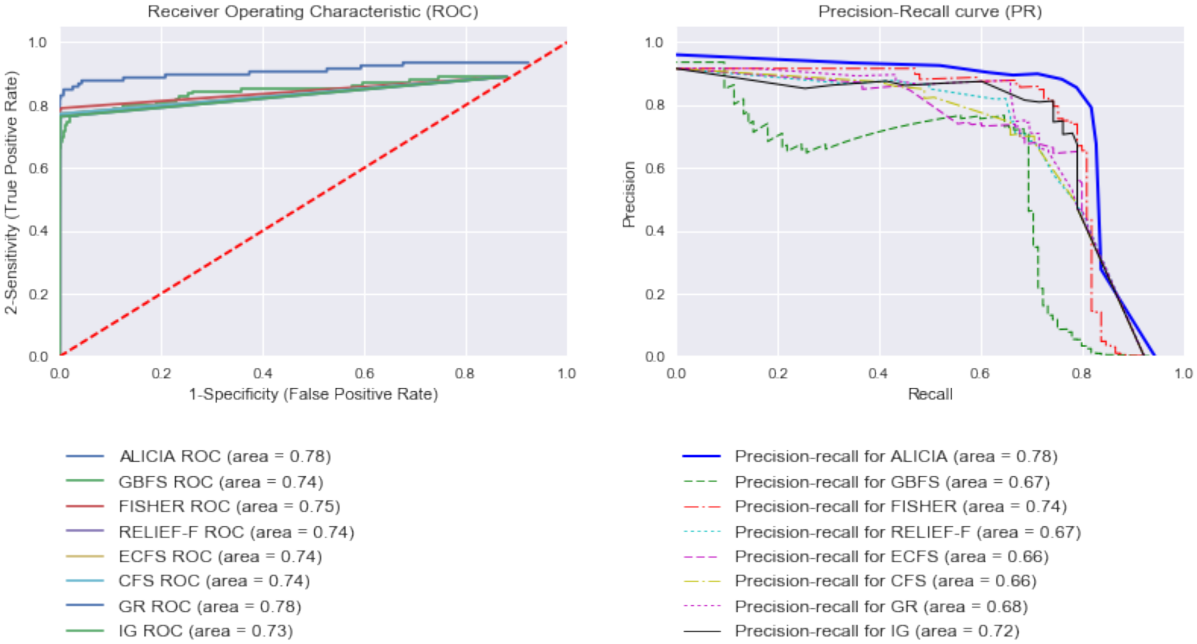

Figure 17 – The same curve shown in both ROC and PR space, for *TFM-2010* dataset.

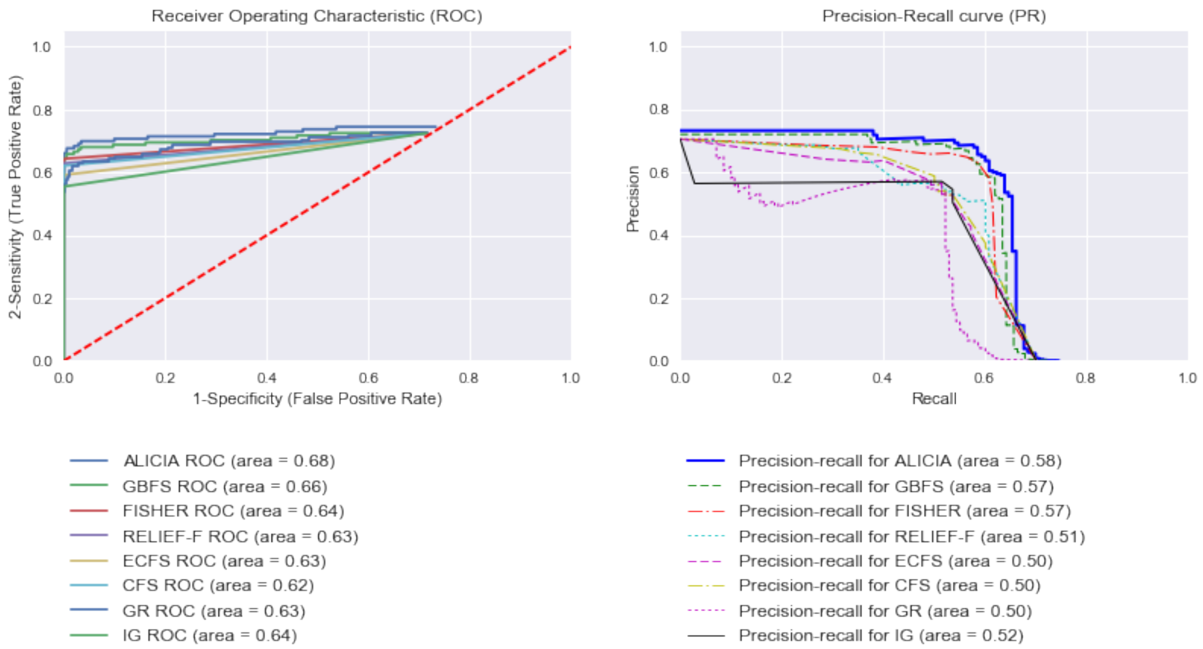

Figure 18 – The same curve shown in both ROC and PR space, for *TFM-2011* dataset.

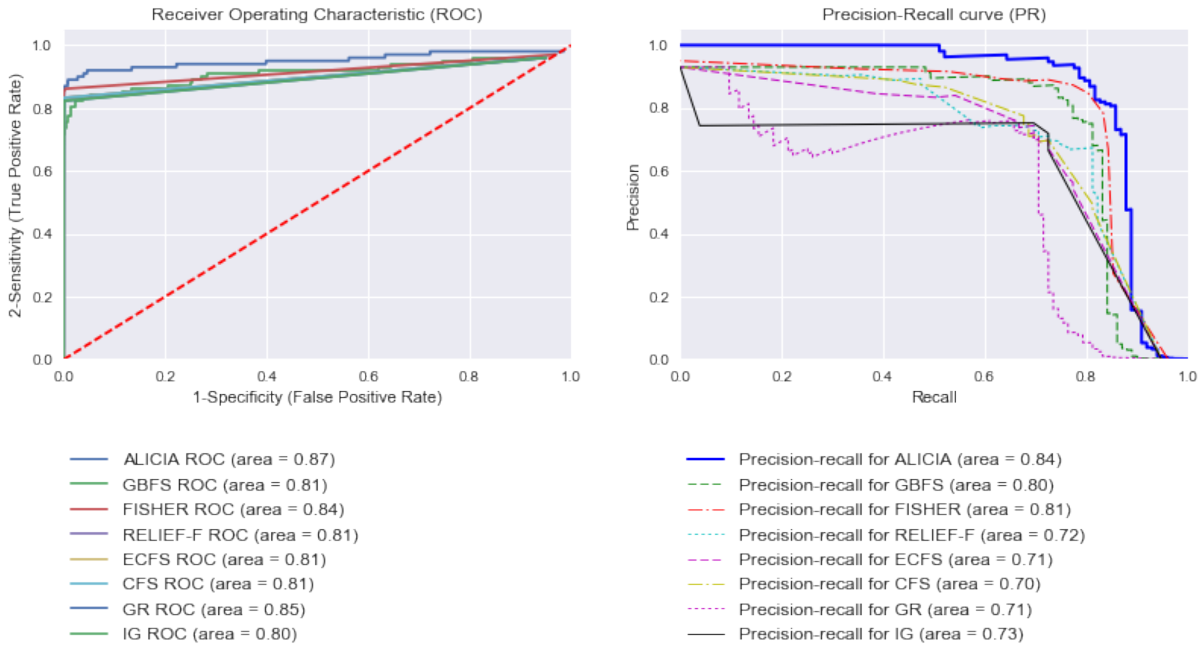

Figure 19 – The same curve shown in both ROC and PR space, for *DS-CreditCard* dataset.

6 CONCLUSION

In this work we propose a feature selection method, ALICIA, that leverages association rules and propositional logic with a novel graph centrality measure, FTI, to improve the detection of potential tax fraudsters. In order to assess the quality of the results we conduct an extensive experimental evaluation, where we compare ALICIA with eight other well-established methods, namely, information gain, gain ratio, correlation-based feature selection, feature Selection via eigenvector centrality, RELIEF-F, Fisher, and Gradient boosted feature selection (GBFS). The evaluation is conducted by considering three real-world datasets provided by the treasury office of the state of Ceará (SEFAZ-CE), and an SVM-based classifier. From the results we first observe that ALICIA consistently outperforms its competitor, achieving best F-Measure scores for *TFM-2009* (71.71%), *TFM-2010* (72.91%), *TFM-2011* (76.88%) and *DS-creditcard* (89.13%).Thus proving its superiority in providing the most appropriate subset of binary features, with low or non-linear correlation between themselves and possibly non informative for the target class, to improve potential tax fraudsters classification.

We also demonstrate how well our strategy is able to generalize in fraud domain using credit card dataset provided by Kaggle with an XGB classifier, achieving F-measure scores up to 89.13%, supplanting its competitors.

While the literature has shown no clear superiority of any particular feature selection method, some feature selection methods are more suitable than other according to the characteristics of the dataset. Correlation is a well-known similarity measure between two random variables. If two random variables are linearly dependent, then their correlation coefficient is ± 1. From the correlation measure between the random variables in section 4.1.1 we demonstrate that random variables are not linearly dependent, in all dataset *TFM-2009*, *TFM-2010*, *TFM-2011* and *DS-creditcard*. For this reason, our method surpasses the traditional feature selection methods - it captures non linear correlation between features in a binary dataset for tax fraudster classification problem.

As a future line of further research, we plan to analyze how our method performs with different propositional logic argument forms, as: conjunction, addition and simplification. We also plan to analyze how does the association patterns change over time, what are the factors that drive the associations, and how is the association between two nodes affected by other nodes. The problem we want to tackle here is to predict the likelihood of a future association between two nodes, knowing that there is no association between the nodes in the current state of the graph. This problem is commonly known as the Link Prediction problem (LIBEN-NOWELL; KLEINBERG, 2007).

Finally, we also plan to work on a distributed version of ALICIA, as centrality measures can be implemented by means of the Map Reduce paradigm (KANG *et al.*, 2011; LERMAN *et al.*, 2010).

REFERENCES

AGRAWAL, R.; SRIKANT, R. Fast algorithms for mining association rules in large databases. In: **Proceedings of the 20th International Conference on Very Large Data Bases**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1994. (VLDB '94), p. 487–499. ISBN 1-55860-153-8.

AMALDI, E.; KANN, V. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. **Theoretical Computer Science**, Elsevier, v. 209, n. 1, p. 237–260, 1998.

ANDREI, A. L.; COMER, K.; KOEHLER, M. An agent-based model of network effects on tax compliance and evasion. **Journal of Economic Psychology**, Elsevier, v. 40, p. 119–133, 2014.

BARRAT, A.; BARTHELEMY, M.; PASTOR-SATORRAS, R.; VESPIGNANI, A. The architecture of complex weighted networks. **Proceedings of the National Academy of Sciences of the United States of America**, National Acad Sciences, v. 101, n. 11, p. 3747–3752, 2004.

BELHADJI, E. B.; DIONNE, G.; TARKHANI, F. A model for the detection of insurance fraud. **The Geneva Papers on Risk and Insurance Issues and Practice**, Springer, v. 25, n. 4, p. 517–538, 2000.

BHATTACHARYYA, S.; JHA, S.; THARAKUNNEL, K.; WESTLAND, J. C. Data mining for credit card fraud: A comparative study. **Decision Support Systems**, Elsevier, v. 50, n. 3, p. 602–613, 2011.

BONACICH, P. Power and centrality: A family of measures. **American journal of sociology**, University of Chicago Press, v. 92, n. 5, p. 1170–1182, 1987.

BONEV, B.; ESCOLANO, F.; GIORGI, D.; BIASOTTI, S. Information-theoretic selection of high-dimensional spectral features for structural recognition. **Computer Vision and Image Understanding**, Elsevier, v. 117, n. 3, p. 214–228, 2013.

BORGATTI, S. P.; MEHRA, A.; BRASS, D. J.; LABIANCA, G. Network analysis in the social sciences. **science**, American Association for the Advancement of Science, v. 323, n. 5916, p. 892–895, 2009.

BREIMAN, L.; FRIEDMAN, J.; STONE, C. J.; OLSHEN, R. A. **Classification and regression trees**. [S.l.]: CRC press, 1984.

CHANDRASHEKAR, G.; SAHIN, F. A survey on feature selection methods. **Computers & Electrical Engineering**, Elsevier, v. 40, n. 1, p. 16–28, 2014.

COHEN, P.; WEST, S. G.; AIKEN, L. S. **Applied multiple regression/correlation analysis for the behavioral sciences**. [S.l.]: Psychology Press, 2014.

CTE. **Tax Evasion in Brazil**. [S.l.], 2019 (Retrieved June 15, 2019). http://www.cte.fazenda.gov.br/portal/.

D'AGOSTINO, R.; PEARSON, E. S. Tests for departure from normality. empirical results for the distributions of b 2 and b. **Biometrika**, Oxford University Press, v. 60, n. 3, p. 613–622, 1973.

D'AGOSTINO, R. B. An omnibus test of normality for moderate and large size samples. **Biometrika**, Oxford University Press, v. 58, n. 2, p. 341–348, 1971.

DASH, M.; LIU, H. Feature selection for classification. **Intelligent data analysis**, Elsevier, v. 1, n. 1-4, p. 131–156, 1997.

DEVIJVER, P.; KITTLER, J. **Pattern Recognition, A statistical approach**. London: Prentice-Hall International, 1982.

DOREIAN, P.; BATAGELJ, V.; FERLIGOJ, A. **Generalized blockmodeling**. [S.l.]: Cambridge university press, 2005. v. 25.

FAGIN, R. Functional dependencies in a relational database and propositional logic. **IBM Journal of research and development**, IBM, v. 21, n. 6, p. 534–544, 1977.

FAWCETT, T. An introduction to roc analysis. **Pattern recognition letters**, Elsevier, v. 27, n. 8, p. 861–874, 2006.

FAWCETT, T.; PROVOST, F. Adaptive fraud detection. **Data mining and knowledge discovery**, Springer, v. 1, n. 3, p. 291–316, 1997.

FREEMAN, L. C. A set of measures of centrality based on betweenness. **Sociometry**, JSTOR, p. 35–41, 1977.

FREEMAN, L. C. Centrality in social networks conceptual clarification. **Social networks**, Elsevier, v. 1, n. 3, p. 215–239, 1978.

GLANCY, F. H.; YADAV, S. B. A computational model for financial reporting fraud detection. **Decision Support Systems**, Elsevier, v. 50, n. 3, p. 595–601, 2011.

GU, Q.; LI, Z.; HAN, J. Generalized fisher score for feature selection. **arXiv preprint arXiv:1202.3725**, 2012.

GUYON, I.; ELISSEEFF, A. An introduction to variable and feature selection. **The Journal of Machine Learning Research**, JMLR. org, v. 3, p. 1157–1182, 2003.

HALL, M. A. **Correlation-based feature selection for machine learning**. Tese (Doutorado) — The University of Waikato, 1999.

HAN, J.; KAMBER, M. Data mining: Concepts and techniques, 2nd editionmorgan kaufmann publishers. **San Francisco, CA, USA**, 2006.

HANLEY, J. A.; MCNEIL, B. J. The meaning and use of the area under a receiver operating characteristic (roc) curve. **Radiology**, v. 143, n. 1, p. 29–36, 1982.

HASHIMZADE, N.; MYLES, G. D.; PAGE, F.; RABLEN, M. D. The use of agent-based modelling to investigate tax compliance. **Economics of Governance**, Springer, v. 16, n. 2, p. 143–164, 2015.

HEGEL, G. W. F. **Science of logic**. [S.l.]: Routledge, 2014.

KANG, U.; PAPADIMITRIOU, S.; SUN, J.; TONG, H. Centralities in large networks: Algorithms and observations. In: SIAM. **Proceedings of the 2011 SIAM International Conference on Data Mining**. [S.l.], 2011. p. 119–130.

KARATZOGLOU, A.; SMOLA, A.; HORNIK, K.; ZEILEIS, A. kernlab – an S4 package for kernel methods in R. **Journal of Statistical Software**, v. 11, n. 9, p. 1–20, 2004. Disponível em: <http://www.jstatsoft.org/v11/i09/>.

KAREGOWDA, A. G.; MANJUNATH, A.; JAYARAM, M. Comparative study of attribute selection using gain ratio and correlation based feature selection. **International Journal of Information Technology and Knowledge Management**, v. 2, n. 2, p. 271–277, 2010.

KIM, K.; CHOI, Y.; PARK, J. Pricing fraud detection in online shopping malls using a finite mixture model. **Electronic Commerce Research and Applications**, Elsevier, v. 12, n. 3, p. 195–207, 2013.

KIRA, K.; RENDELL, L. A. The feature selection problem: Traditional methods and a new algorithm. In: **Aaai**. [S.l.: s.n.], 1992. v. 2, p. 129–134.

KIRA, K.; RENDELL, L. A. A practical approach to feature selection. In: **Proceedings of the ninth international workshop on Machine learning**. [S.l.: s.n.], 1992. p. 249–256.

KIRKOS, E.; SPATHIS, C.; MANOLOPOULOS, Y. Data mining techniques for the detection of fraudulent financial statements. **Expert Systems with Applications**, Elsevier, v. 32, n. 4, p. 995–1003, 2007.

KITTLER, J. Feature selection and extraction. **Handbook of pattern recognition and image processing**, p. 59–83, 1986.

KOHAVI, R.; JOHN, G. H. Wrappers for feature subset selection. **Artificial intelligence**, Elsevier, v. 97, n. 1, p. 273–324, 1997.

KOHAVI, R. *et al.* A study of cross-validation and bootstrap for accuracy estimation and model selection. In: **Ijcai**. [S.l.: s.n.], 1995. v. 14, n. 2, p. 1137–1145.

KONONENKO, I. Estimating attributes: analysis and extensions of relief. In: SPRINGER. **European conference on machine learning**. [S.l.], 1994. p. 171–182.

KOROBOW, A.; JOHNSON, C.; AXTELL, R. An agent–based model of tax compliance with social networks. **National Tax Journal**, JSTOR, p. 589–610, 2007.

KOTSIANTIS, S.; KOUMANAKOS, E.; TZELEPIS, D.; TAMPAKAS, V. Forecasting fraudulent financial statements using data mining. **International Journal of Computational Intelligence**, v. 3, n. 2, p. 104–110, 2006.

KRAJICEK, J.; KRAJÍČEK, J. *et al.* **Bounded arithmetic, propositional logic and complexity theory**. [S.l.]: Cambridge University Press, 1995.

LERMAN, K.; GHOSH, R.; KANG, J. H. Centrality metric for dynamic networks. In: ACM. **Proceedings of the Eighth Workshop on Mining and Learning with Graphs**. [S.l.], 2010. p. 70–77.

LI, J.; HUANG, K.-Y.; JIN, J.; SHI, J. A survey on statistical methods for health care fraud detection. **Health care management science**, Springer, v. 11, n. 3, p. 275–287, 2008.

LI, S.-H.; YEN, D. C.; LU, W.-H.; WANG, C. Identifying the signs of fraudulent accounts using data mining techniques. **Computers in Human Behavior**, Elsevier, v. 28, n. 3, p. 1002–1013, 2012.

LIBEN-NOWELL, D.; KLEINBERG, J. The link-prediction problem for social networks. **Journal of the American society for information science and technology**, Wiley Online Library, v. 58, n. 7, p. 1019–1031, 2007.

LIU, H.; MOTODA, H. **Computational methods of feature selection**. [S.l.]: CRC Press, 2007.

LIU, H.; MOTODA, H. **Feature selection for knowledge discovery and data mining**. [S.l.]: Springer Science & Business Media, 2012. v. 454.

MATOS, T.; MACEDO, J. A. F. de; MONTEIRO, J. M. An empirical method for discovering tax fraudsters: A real case study of brazilian fiscal evasion. In: ACM. **Proceedings of the 19th International Database Engineering & Applications Symposium**. [S.l.], 2015. p. 41–48.

MATOS, T.; MACÊDO, J. A. F. de; MONTEIRO, J. M.; LETTICH, F. An accurate tax fraud classifier with feature selection based on complex network node centrality measure. In: **ICEIS 2017 - Proceedings of the 19th International Conference on Enterprise Information Systems, Volume 1, Porto, Portugal, April 26-29, 2017**. [s.n.], 2017. p. 145–151. Disponível em: <https://doi.org/10.5220/0006335501450151>.

MDFE. **Tax Evasion in Brazil**. [S.l.], 2019 (Retrieved June 15, 2019). http://sped.rfb.gov.br/projeto/show/1312.

MORADI, P.; ROSTAMI, M. A graph theoretic approach for unsupervised feature selection. **Engineering Applications of Artificial Intelligence**, Elsevier, v. 44, p. 33–45, 2015.

NFE. **Tax Evasion in Brazil**. [S.l.], 2019 (Retrieved June 15, 2019). http://www.nfe.fazenda.gov.br/portal/principal.aspx.

NGAI, E.; HU, Y.; WONG, Y.; CHEN, Y.; SUN, X. The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. **Decision Support Systems**, Elsevier, v. 50, n. 3, p. 559–569, 2011.

OPSAHL, T.; PANZARASA, P. Clustering in weighted networks. **Social networks**, Elsevier, v. 31, n. 2, p. 155–163, 2009.

PHUA, C.; LEE, V.; SMITH, K.; GAYLER, R. A comprehensive survey of data mining-based fraud detection research. **arXiv preprint arXiv:1009.6119**, 2010.

POZZOLO, A. D.; CAELEN, O.; JOHNSON, R. A.; BONTEMPI, G. Calibrating probability with undersampling for unbalanced classification. In: IEEE. **Computational Intelligence, 2015 IEEE Symposium Series on**. [S.l.], 2015. p. 159–166.

PROVOST, F. J.; FAWCETT, T.; KOHAVI, R. *et al.* The case against accuracy estimation for comparing induction algorithms. In: **ICML**. [S.l.: s.n.], 1998. v. 98, p. 445–453.

RAVISANKAR, P.; RAVI, V.; RAO, G. R.; BOSE, I. Detection of financial statement fraud and feature selection using data mining techniques. **Decision Support Systems**, Elsevier, v. 50, n. 2, p. 491–500, 2011.

REFERENCE, P. A. **XGBoost - Python API reference**. 2018 (Retrieved June 23, 2018). Disponível em: <https://xgboost.readthedocs.io/en/latest/python/python_api.html>.

RICHHARIYA, P.; SINGH, P. K.; DUNEJA, E.; BITS, B.; SOFTWARES, I. A survey on financial fraud detection methodologies. **International Journal of Computer Applications**, International Journal of Computer Applications, 244 5 th Avenue,# 1526, New York, NY 10001, USA India, v. 45, n. 22, 2012.

ROFFO, G.; MELZI, S. Features selection via eigenvector centrality. **Proceedings of New Frontiers in Mining Complex Patterns (NFMCP 2016)(Oct 2016)**, 2016.

ROFFO, G.; MELZI, S.; CASTELLANI, U.; VINCIARELLI, A. Infinite latent feature selection: A probabilistic latent graph-based ranking approach. In: **Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2017.

ROMANSKI, P. Fselector: Selecting attributes. 2009. **R package version 0.18, URL http://CRAN. R-project. org/package= FSelector. Accessed**, v. 30, 2016.

RUIZ, R.; SANTOS, J. C. R.; AGUILAR-RUIZ, J. S. Heuristic search over a ranking for feature selection. In: SPRINGER. **IWANN**. [S.l.], 2005. p. 742–749.

SABIDUSSI, G. The centrality index of a graph. **Psychometrika**, Springer, v. 31, n. 4, p. 581–603, 1966.

SAITO, T.; REHMSMEIER, M. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. **PloS one**, Public Library of Science, v. 10, n. 3, p. e0118432, 2015.

SÁNCHEZ, D.; VILA, M.; CERDA, L.; SERRANO, J.-M. Association rules applied to credit card fraud detection. **Expert Systems with Applications**, Elsevier, v. 36, n. 2, p. 3630–3640, 2009.

SPED. **Tax Evasion in Brazil**. [S.l.], 2019 (Retrieved June 15, 2019). http://sped.rfb.gov.br.

WIKIPEDIA. **Betweenness centrality — Wikipedia, The Free Encyclopedia**. 2017. <http://en.wikipedia.org/w/index.php?title=Betweenness%20centrality&oldid=777459351>. [Online; accessed 05-May-2017].

XU, E. **Gradient Boosted Feature Selection code**. [S.l.], 2018 (Retrieved April 19, 2018). http://www.cse.wustl.edu/ xuzx/research/code/code.html.

YANG, W.-S.; HWANG, S.-Y. A process-mining framework for the detection of healthcare fraud and abuse. **Expert Systems with Applications**, Elsevier, v. 31, n. 1, p. 56–68, 2006.

YU, L.; LIU, H. Efficient feature selection via analysis of relevance and redundancy. **The Journal of Machine Learning Research**, JMLR. org, v. 5, p. 1205–1224, 2004.

ZAFFALON, M.; HUTTER, M. Robust feature selection using distributions of mutual information. In: **Proceedings of the 18th International Conference on Uncertainty in Artificial Intelligence (UAI-2002)**. [S.l.: s.n.], 2002. p. 577–584.

ZHANG, X.; TOKOGLU, F.; NEGISHI, M.; ARORA, J.; WINSTANLEY, S.; SPENCER, D. D.; CONSTABLE, R. T. Social network theory applied to resting-state fmri connectivity data in the identification of epilepsy networks with iterative feature selection. **Journal of neuroscience methods**, Elsevier, v. 199, n. 1, p. 129–139, 2011.

ZHANG, Y.; BIAN, J.; ZHU, W. Trust fraud: A crucial challenge for china's e-commerce market. **Electronic Commerce Research and Applications**, Elsevier, v. 12, n. 5, p. 299–308, 2013.

ANNEX A – SEFAZ-CE: DATA ACCESS PERMISSION

GOVERNO DO
ESTADO DO CEARÁ
Secretaria da Fazenda

Processo nº: 05022171/2019
Interessado: Raimundo Tales B R Matos
Assunto: Solicitação de Acesso a Dados para Utilização em Pesquisa Científica

Ao Servidor Raimundo Tales B R Matos,

DESPACHO

Em atendimento à solicitação proferida pelo servidor, com o intuito de subsidiar pesquisa científica, autorizamos o acesso às informações constantes nas bases de dados da Sefaz, sem identificação dos contribuintes ou setores, devendo ser observado o sigilo fiscal, que garante a preservação dos referidos dados.

Fortaleza, 17 de junho de 2019

Atenciosamente,

Elizangela Amaral de M. Bezerra
Auditora Fiscal do Estado do Ceará
SEFAZ-CE Mat - 497593.1 2

Elizângela Amaral de Moura Bezerra

SUPERVISORA DA CEARP