T.C. DOĞUŞ UNIVERSITY

INSTITUTE OF SCIENCE AND TECHNOLOGY

ENGINEERING AND TECHNOLOGY MANAGEMENT DEPARTMENT

DETECTION OF DEBIT CARD FRAUD THROUGH RANDOM FOREST

Master Thesis

Kasım AKSOY

201199013

Advisor: Mesut KUMRU

İSTANBUL, AUGUST 2017

T.C. DOĞUŞ UNIVERSITY

INSTITUTE OF SCIENCE AND TECHNOLOGY

ENGINEERING AND TECHNOLOGY MANAGEMENT DEPARTMENT

DETECTION OF DEBIT CARD FRAUD THROUGH RANDOM FOREST

Master Thesis

Kasım AKSOY
201199013

Advisor

Prof. Dr. Mesut KUMRU

Jury
Assoc. Prof. Dr. Ali Fuat ALKAYA
Assist. Prof. Dr. Kıvanç ONAN

İSTANBUL, AUGUST 2017

PREFACE

First of all, I would like to thank to my advisor Professor Mesut KUMRU for his patience and academic guidance. My director and my colleagues have encouraged me to complete my thesis. Finally, I want to express my sincere gratitude to my family for their support.

Istanbul, August 2017 Kasım AKSOY

ÖZET

Günlük hayatımızda en sık kullandığımız finansal araçlardan olan ATM’ler, kullanıma geçtikleri tarihten itibaren aynı sıklıkla dolandırıcıların hedefi olagelmiştir. Özellikle manyetik bant kullanılarak üretilen ATM kartlarının (debit kart) güvenlik açıkları dolandırıcılar tarafından bir fırsat olarak görülmüştür. Bu güvenlik açıkları istismar edilerek gerçekleştirilen kart kopyalama vakaları sonucu müşteri hesaplarından önemli miktarda dolandırıcılık yapılmıştır. Bu tez çalışmasında, bir bankaya ait ATM nakit çekim işlem verisi kullanılarak ATM kartı dolandırıcılıklarının tespit edilmesi için bir model ortaya konulmuştur. Öncelikle ATM nakit çekim işlem veri setinde dolandırıcılık tespiti ile ilgili olabileceği düşünülen işlem değişkenleri tespit edilmiştir. Akabinde, bu değişkenler üzerinden literatürde dolandırıcılık tespitinde kullanılabileceği belirtilen RFM (Recency-Yakınlık, Frequency-Sıklık, Monetary-Parasal büyüklük) değişkenleri hesaplanmıştır. İkinci adımda RFM değişkenleri ve nakit çekim işlem değişkenleri kullanılarak rastgele orman algoritması ile bir sınıflandırma modeli oluşturulmuştur. Üçüncü olarak oluşturulan sınıflandırma modeli algoritmanın farklı parametreleriyle test edilmiştir. Çalışmanın sonuç kısmında hazırlanan modelin sonuçları tartışılmış ve pratik uygulamalar ışığında gerçek zamanlı bir ATM kartı dolandırıcılık tespit sistemi kurulması konusunda bazı değerlendirmeler sunulmuştur.

SUMMARY

As one of the most frequently used financial tools in our life, ATMs have become a target for fraudsters in the same frequency. Particularly, security vulnerabilities of debit cards, which are generally produced by using magnetic stripes, was seen as an opportunity for fraud. As a result of exploiting those security vulnerabilities, important amounts have been fraudulently withdrawn from customer accounts. In this thesis, a data mining model was established for detection of debit card fraud through debit card transaction data of a bank. Firstly, transaction variables were defined in the ATM cash withdrawal dataset with consideration of their relevance in the debit card fraud detection. Consequently, behavioral RFM (Recency, Frequency, Monetary) variables, which are suggested as relevant in debit card fraud detection literature, were calculated based on those transaction variables. Secondly, several experiments were made through the classification model created by random forest algorithm by changing algorithm parameters. In the concluding remarks, the results of the established model were summarized and, considering practical implementations, some assessments regarding a real-time debit card fraud detection system were made.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

Since their inception in 1960's, it is not an exaggeration to say that ATMs have revolutionized the way we reach money. They made it possible to withdraw money without limitations of working hours. Albeit slowed in some regions of the world due to saturation, the large-scale adoption of ATMs continues with enhanced features derived from technological innovations.

As with any financial instrument, ATMs were subject to the wrongdoings of fraudsters since their launch in the market. Through various methods including skimming magnetic stripe or capturing the card, fraudsters not only cause financial losses for the bank, but also endanger customer confidence.

In this thesis, we aim to establish a model for the detection of fraudulent cash withdrawals from ATMs. In the first section, we gave a brief definition of fraud and provided concise explanations regarding frequent types of frauds. We also gave more detail for plastic card fraud.

In the second section, various data mining methods utilized in the fraud detection are summarized and important considerations in their usage are delineated. The challenges of detecting fraud through data mining or data analysis were also explained. In the second part of that section, a review of literature with regard to the usage of data mining techniques in fraud detection was undertaken.

In the last section, the debit card fraud data of a bank for a given period is examined and relevant and available transaction variables are defined. Based on those transaction variables, new behavioral variables were devised based on RFM (recency, frequency and monetary) feature extraction/aggregation. Those variables were computed for our dataset. Consequently, our dataset was divided between training and test portions. Finally, a model was established through random forest algorithm and several experiments were designed and implemented for understanding the impact of parameter and stratified sampling changes on the model's prediction performance.

In our concluding remarks, we discussed the results of our experiments and limitations of the produced model. Recommendations were also provided for future research. Finally, based on the established model and practical requirements, some assessments were made regarding a real-time debit card fraud detection system.

1. UNDERSTANDING PLASTIC CARD FRAUD

Used in such diverse fields as credit card, insurance, plagiarism, fraud is a popular phenomenon which takes many forms in different fields. In this section, we will give some of the definitions of fraud and introduce various forms of fraud with special emphasis on plastic card fraud.

1.1. Definitions of Fraud

Many definitions of fraud were made to emphasize different dimensions of the misconducts. In this section, we will mention some of the prominent fraud definitions and explore the similarities in the definitions. According to the Institute of Internal Auditors, IAA, fraud is

“Any illegal acts characterized by deceit, concealment, or violation of trust. These acts are not dependent upon the application of threat of violence or of physical force. Frauds are perpetrated by individuals and organizations to obtain money, property, or services; to avoid payment or loss of services; or to secure personal or business advantage “(Institute of Internal Auditors, 2016).

According to the Association of Certified Fraud Examiners, ACFE, fraud includes “any intentional or deliberate act to deprive another of property or money by guile, deception, or other unfair means” (ACFE, 2016). Violations can range from asset misappropriation, fraudulent statements and corruption over pilferage and petty theft, false overtime and using company property for personal benefit to payroll and sick time abuses (Jans et al., 2006).

According to Baesens et al. (2015), “Fraud is an uncommon, well considered, imperceptibly concealed, time-evolving and often carefully organized crime which appears in many types and forms.” (Baesens et al., 2015)

According to Vona, fraud is “Acts committed on the organization or by the organization or for the organization. The acts are committed by an internal or external source and are intentional and concealed. The acts are typically illegal or denote wrongdoing, such as in the cases of: financial misstatement, policy violation, ethical lapse, or a perception issue. The acts cause a loss of company funds, company value, or company reputation, or any unauthorized benefit whether received personally or by others” (Vona, 2008).

The significant points of definitions may be listed as follows:

a. Fraud causes loss for the organization and/or lead undeserved benefit

b. Contains deceit or concealment, not violence
c. Intentional, not erroneous
d. May have many forms
e. Evolves in time, thus, it is similar to cat and mouse game (Baesens et al., 2015)
f. Involves illegal actions or wrongdoings

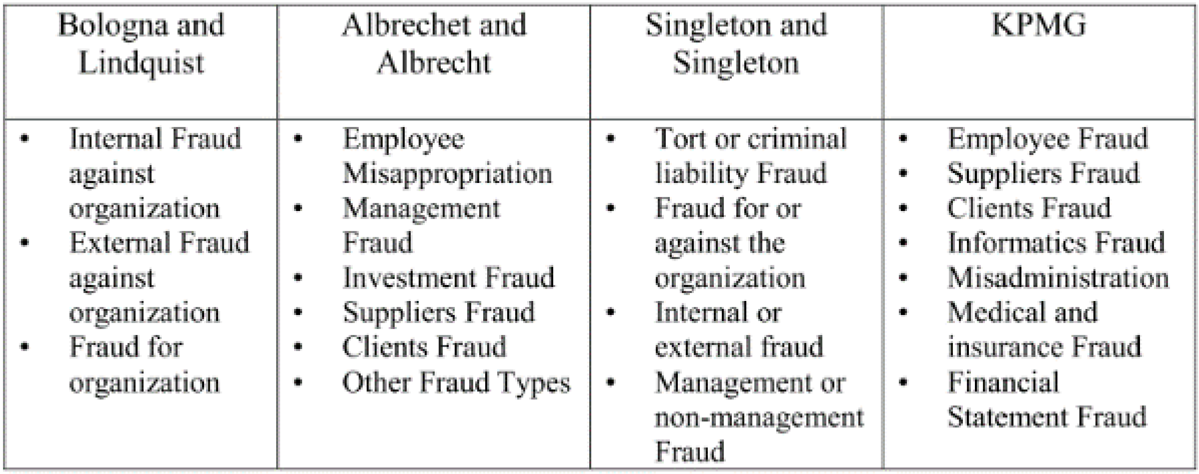
In several contexts, other concepts may have used as synonyms of fraud. For instance (PwC, 2016) introduces economic crimes and (Ernst & Young, 2016) introduces corporate misconduct as quite similar to fraud definition. In this thesis, we have used fraud as the overarching concept for all such similar terms.

1.2. Fraud Classification

Disguising in many different misconducts, a nonexhaustive list of fraud includes credit card fraud, insurance fraud, corruption, counterfeit, product warranty fraud, healthcare fraud, telecommunications fraud, money laundering, click fraud, identity theft, tax evasion, plagiarism and embezzlement (Baesens et al., 2015).

According to ACFE, fraud can be realized against an organization or against an individual. Fraud against an organization can be committed by internal actors like employees, managers, officers or owners. On the other hands, external factors like customers, vendors and other parties may also commit fraud against organization (ACFE, 2016). In the following table, some of the fraud classifications made by different authors are given.

Table 1.1. Fraud taxonomies



Bologna and Lindquist	Albrechet and Albrecht	Singleton and Singleton	KPMG
• Internal Fraud against organization • External Fraud against organization • Fraud for organization	• Employee Misappropriation • Management Fraud • Investment Fraud • Suppliers Fraud • Clients Fraud • Other Fraud Types	• Tort or criminal liability Fraud • Fraud for or against the organization • Internal or external fraud • Management or non-management Fraud	• Employee Fraud • Suppliers Fraud • Clients Fraud • Informatics Fraud • Misadministration • Medical and insurance Fraud • Financial Statement Fraud

Source: Sabau (2010)

Internal Fraud, also known as occupational fraud, is defined as “the use of one’s occupation for personal enrichment through the deliberate misuse or misapplication of the organization’s resources or assets”. External fraud, on the other hand, includes many schemes perpetrated by vendors, customers and unknown third parties to unlawfully gather any kind of resources (ACFE, 2016). In this study, we will focus on debit card fraud, which is mainly an external fraud type.

1.3. Cost of Fraud

As a global problem, fraud inflicts huge losses to the victims, which includes individuals, enterprises or the government. Hence, the cost of fraud affects all society. Although it is impossible to measure the amount of losses precisely, some recent numbers suggest that

- A typical organization loses 5 percent of its revenue to fraud each year.
- The total cost of insurance fraud (excluding medical insurance) in the US is estimated to be more than 40 billion USD per year.
- The cost of fraud in the UK is 73 billion pound each year
- Credit card companies lose 7 cents for each 100 dollar transaction (Baesens et al., 2015).

The reports generally mention an increasing fraud tendency in many fields. Part of this increase can be attributed to the development of information technologies, particularly internet and mobile technologies.

In terms of losses associated with deposit account fraud, Deposit Account Survey of American Banking Association gives us some clues. According to the report, the cost of demand deposit account (DDA) fraud in the USA at 2014 is about USD 1.9 billion, with a distribution of debit card fraud (66%), check fraud (32%) and online banking (2%) (Kenneally et al., 2016). According to the study, the fraud amount has increased since the previous survey which is held at 2012. The survey also suggests that 74% of respondents think that the increase in fraud cases is linked to increase in fraud attempts.

Beside financial losses incurred by companies, fraud also have detrimental impacts on societies (Vona, 2008). The costs of fraud are directly reflected to the prices charged by the firms. Another dimension of the negative impacts of fraud is related with the frequent linkage between fraudulent transactions and illicit activities such as drug trafficking and organized crime (West & Bhattacharya, 2016).

1.4. Plastic Card Fraud

In our thesis, we will establish a model for debit card fraud detection. Debit cards and credit cards (plastic cards) are issued by the banks to enable cash withdrawal on 7/24 basis from ATMs and making purchases through POS (point of sale) devices in online or on-site merchants. In this section, we will briefly explain the operations of debit cards and credit cards. Later, we will explore most common fraud schemes encountered in plastic cards.

1.4.1. Debit Cards, Credit Cards and ATMs

A debit card or bank card is a plastic payment card issued by a financial institution to enable customers to withdraw cash from their accounts and to make purchases. Any purchase or cash withdrawal made through debit card is debited to the customer account. Hence, in principle, it is not possible to use the card more than the available balance of the customer account. There are various types of debit cards including online (EFTPOS), offline and electronic purse card. Online debit cards directly communicate with the bank to check account balance. The payments made via offline debit cards, on the other hand, are reflected to bank account within a few days. Electronic purses have chips which store the account balance info and hence they need no network connectivity in POS devices (Wikipedia, 2017c).

In general, payment cards that have the ability to withdraw cash from ATMs are named as ATM cards. In that sense, performing basic banking operations like withdrawal in ATM is another important function of debit cards. Credit cards may also be used to withdraw money from bank accounts (Wikipedia, 2017a).

A credit card is a type of payment card issued to cardholders to enable them to pay a merchant for goods and services. This service is based on the cardholder's promise to the card issuer to pay them for the purchased amounts. The issuers of credit cards generally extend a line of credit to the customer for purchases or cash advances. A charge card, on the other hand, requires the balance to be repaid in full each month (Wikipedia credit card).

Mainly, credit cards and debit cards may be used in three different ways: ATM, POS- Card Present (CP) and POS- Card Not Present (CNP) transactions. POS-CP transactions are conducted on physical POS devices in the presence of the card. POS-CNP transactions, on the other hand, represent the mail order purchases made on via either telephone or internet (Krivko, 2010).

An Automated Teller Machine (ATM) is a banking channel for performing basic banking transactions without branch interaction. Since their inception in 1960's, they gained huge popularity. As of now, more than 3.5 million ATM's were installed around the globe and the number is rising. In some countries, banks create ATM networks that enable cash withdrawal from ATMs of any affiliated bank included in the network (Wikipedia, 2017b). Due to their ease of use, cash withdrawal from ATM has now preceded the withdrawal from branch particularly for small amounts. An ATM is connected to the databases of the bank and activated by a customer to withdraw cash or to make other banking transactions. It is essentially a computer with a keypad and screen to access accounts and make transactions (Sharma, 2012).

Although they are originally developed as cash dispensers, ATMs have evolved to include many other bank-related functions (Wikipedia, 2017b):

- Paying routine bills, fees, and taxes (utilities, phone bills, social security, legal fees, income taxes, etc.)
- Printing bank statements
- Updating passbooks
- Cash advances
- Cheque Processing Module
- Paying (in full or partially) the credit balance on a card linked to a specific current account.
- Transferring money between accounts
- Deposit currency recognition, acceptance, and recycling

1.4.2. Realization of ATM Cash Withdrawals

Debit and credit cards can be used to withdraw money from ATMs. ATMs are connected to the databases of the owner bank and, if they are part of an interbank ATM network, they are also connected to the databases of affiliated banks. This connection may be realized through leased lines, which are faster but expensive to operate, or dial-up/DSL connection on a public phone line (Seksaria, 2016).

An ATM has two input devices which are card reader and keypad. Card reader gets the specific account information stored on the magnetic stripe or EMV chip of a debit or credit card. The host processor uses this information to route the transaction to the cardholder's bank. Keypad, along with buttons on the screen, lets the cardholder tell the ATM the type of demanded transaction and required amount, if relevant for type of transaction. In terms of security, keypads are also used for entering the Personal Identification Number (PIN) which is required by the bank to authorize the transaction (Seksaria, 2016).

Debit and credit cards are mainly electronic storage devices. Storage may be carried out by either a magnetic stripe or EMV chip. A magnetic stripe is similar to cassette tape in terms of storing information. The stored information is account number, cardholder name, expiration date and security codes together with card verification value (PIN). Those data except PIN does not change and unprotected for copying. When a payment card with magnetic stripe is swiped through a reader, the information on the magnetic stripe is transmitted to an acquirer company which collects card requests, verifies the request, and makes or guarantees payment to the merchant. In each request, the acquirer checks the merchant ID, card number, expiration date, card limit, and any control related with card usage. Following swipe, the card is returned to the card holder (Toast Inc, 2017).

EMV stands for Europay-Visa-Mastercard, the largest transaction processors in the world, and represents higher security standards for payment cards. EMV enabled devices communicate with a chip in the EMV payment card. That chip contains required information to authenticate the card following a pin is entered or a signature is provided by the card holder to validate his/her authority.

Each time the EMV payment card is inserted into the compatible card terminal, the chip communicates with the EMV terminal and validates the card. In an EMV transaction, unlike magnetic stripe transaction, payment card remains in the terminal until the transaction is complete. The card and terminal communicate and agree on the application to run. The cardholder's information is protected by cryptographic keys; thus, the holder's data is protected. Due to those features, duplicating EMV card is extremely difficult (Toast Inc, 2017).

1.4.3. Fraud Schemes in ATM Transactions

As with any other monetary transactions, ATM operations have been subject to fraudulent intentions. While some of the schemes are aiming at capturing payment card information, or physically getting the card, others directly target stored money in the ATM.

- **Card Skimming:** Skimming refers to the stealing of the payment card information thus enabling criminals to counterfeit card. Consumers conduct a normal ATM transaction and are usually unable to notice a problem until their account is defrauded. It is the most outstanding global threat, but its size is shrinking due to countermeasures taken such as anti-skimming solutions, EMV technology and contactless ATM functionality (NCR, 2016).
- **Card Trapping:** Trapping is the stealing of the physical card through a device fixed to the ATM. Before initiation of EMV standards, the PIN does not need to be compromised while trapping the card.
- **Transaction Reversal Fraud:** TRF involves deliberate creation of an error that makes it appear that the cash had not been dispensed. Due to error, the account is re-credited the amount and the actually 'withdrawn' money is taken by fraudsters. This type of fraud could be a type of physical grabbing or a corruption of the transaction message.
- **Cash Trapping:** This type of frauds generally realized with low values. The fraudster uses a device to trap the dispensed cash physically when a customer withdraw cash. When the customer foregoes and leaves ATM location, criminals come to collect trapped cash.
- **Physical Attacks:** This category is related to any attempt to rob the cash in the ATM safe. Methods of physical attacks include solid and gas explosives, as well as removing the ATM from the site and then using other methods to gain access to the safe.
- **Logical Attacks:** Emerging as a significant financial crime, logical attacks have the potential to cause large amounts of losses and they are an emerging type of financial crime in ATMs. Electronic devices, or malicious software can be used in the crime. Through those tools, criminals aim at controlling the ATM machine dispenser to withdraw money. In a different variant of attack, criminals aim at intercepting the card and PIN data of the customers. Such data can then be used in fraudulent transactions. Intercepting the ATM software may be named as man in the middle attack (NCR, 2016).

Following the retrieval of debit card information, or debit card itself depending on the applied scheme, fraudsters try to withdraw all money as early as possible. However, many banks posed daily limits regarding withdrawals or POS transactions for debit cards. For this reason, fraudsters could generally withdraw the money through a time period in line with determined limits. The customer may detect wrongdoing in the first fraudulent transaction, but it may take days or weeks depending on the circumstances and whole balance of the customer may be fraudulently withdrawn.

Unlike credit card frauds, for which merchants are held liable in most cases, the cost of debit card frauds is generally paid by the issuing bank. For this reason, timely detection of debit card frauds is important to limit losses arising from fraud. This study aims to use various data mining algorithms for early detection of debit card frauds.

2. DETECTING FRAUD THROUGH DATA MINING

2.1. Rule-Based (Expert-Based) Fraud Detection

Rule based fraud detection can be defined as specifying rules based on the experience, intuition and business or domain knowledge of experts. Examining previously occurred frauds, some rules can be defined for detection of suspicious transactions. A sample rule in a car insurance fraud detection system, which can be considered in that group, is given below.

In a car accident claim, if:

- Amount of claim is above threshold OR
- Severe accident, but no police report OR
- Severe injury, but no doctor report OR
- Claimant has multiple versions of accident OR
- Multiple receipts submitted,

Then flag claim as suspicious (Baesens et al., 2015).

Some of the similar concepts to rule-based fraud detection are fraud schemes (Vona, 2008) and data-driven fraud detection (Albrecht, 2013). Albrecht (2013) suggests that an analysis of possible fraud symptoms would normally produce 50 fraud schemes and 3-5 symptoms for each fraud scheme. That would create some 250 fraud symptoms to analyze in a typical rule-based detection scheme.

Rule-based fraud detection may be understood as the first step in the data analytics maturity. It can be stated that the first detection attempts regarding fraud start with such rules. In fact, they may be very successful to cover the existing loopholes. However, while the fraud detection attempts are being matured, it is required to go beyond the list of rules, due to some apparent disadvantages of rule-based detection particularly in a data-intensive and complex fraud environment:

- It is difficult and expensive to build, maintain and manage rule database. Rules need to be kept up-to-date and rules should only show most suspicious cases due to required manual investigation.

- Since the rules are based on past experience, newly emerging fraud patterns are not automatically flagged.
- Expert-based systems require the inclusion of domain experts for all “input, evaluation and monitoring” stages, hence they are quite labor intensive (Baesens et al., 2015).
- Auditors/investigators are generally under time pressure for fraud detection. Research shows that time pressure may lead less sensitivity to fraud cues, which makes examining so many rules inefficient.
- Auditors/investigators may lack adequate level of fraud scheme information, which is a prerequisite for managing and examining rules (Albrecht & Albrecht, 2013).

2.2. Data Mining

As a result of computerization of our society, vast amounts of data are being collected in all aspects of our daily life including business, science, engineering and medicine. Businesses worldwide are generating huge datasets that include sales transactions, trading records, product specifications and customer feedbacks. Having billions of transactions, telecommunication companies carry insurmountable amount of data every day. Web searches, online communities and social networks produce huge amount of data as well. Furthermore, Internet of Things (IoT), which means communication between physical objects, is expected to explode the data in the databases in the near future. This growing and available data brought the need to uncover valuable information from it and transform it to the organized knowledge, which is the basic attribute of data mining. The definition also emphasizes that dimension: “Data mining is the process of discovering interesting patterns and knowledge from large amounts of data. The data sources can include databases, data warehouses, the Web, other information repositories, or data that are streamed into the system dynamically” (Han et al., 2011).

Data mining is seen as the next stage in the evolution of information technologies after the stage of database systems. The main tenet of the data mining is that “the world is data rich but information poor”, that is we are living an age of rich data, but we are unable to use that data to produce valuable knowledge. For this reason, the synonym of data mining is Knowledge Discovery in Databases (KDD). The term of data mining also reflects the difficulty of the process, because it requires handling so much raw material (data) to reach usable information (Han et al., 2011).

In terms of fraud detection, according to Baesens et al (2015), data mining based fraud-detection methodologies are gaining power over expert-based approaches for various reasons:

a. Precision: Those methods have enhanced detection power. Using voluminous data, they can uncover patterns that is not possible to be captured by human eye. The power of data mining methods in that sense, which is also proven in credit scoring field, provide effective prioritization of suspicious transactions
b. Operational efficiency: The sheer volume of data requires automatic handling of data. On the other hand, in some settings like credit card transactions, it is required to detect the fraudulent transaction within a very short time limit, which is possible with automatic data-mining based fraud detection methodologies.
c. Cost efficiency: Developing a workable rule-based system is both challenging and labor intensive. It is apparent that data-driven detection methods have cost efficiency, but considering the Pareto principle of decreasing returns (Baesens et al., 2015).

Data mining based detection techniques can be classified as supervised learning techniques, unsupervised learning techniques and social network techniques.

Unsupervised learning techniques or descriptive analytics aimed at detecting *anomalies*, which means deviating records. They learn from historical records, but not require them to be labelled as fraud or not. Beside addressing already flagged fraud, which can be defined as outliers, they may also describe suspicious patterns which are previously not identified. They may fail to address transactions in which the fraudster achieves to simulate suspicious transaction to a normal one.

Supervised learning techniques, or predictive analytics try to determine fraudulent transactions using the pattern learned from previous records. Due to their learning from previous labelled data, they may fail to detect new types of frauds. Their need of having labelled data may be difficult to meet in many fraud detection scenarios (Baesens et al., 2015).

Finally, social network analysis learns from network of linked entities to detect fraudulent behavior. It helps discovering cases where fraudsters are in some way linked. It therefore provides an extra source of information which help uncovering particular fraud cases. Social network analysis is the newest tool in fraud detection techniques.

Considering different groups of fraud detection techniques, it is important to note that they should be used as complementary. Organizations should start from rule-based detection, continue with descriptive detection, predictive detection and social network analysis, although the usage may differ according to cases. Improving the utilized fraud detection techniques should be gradual and stepwise to reach satisfactory conclusions (Baesens et al., 2015).

2.2.1. The Process of Fraud Data Mining

Data mining requires carrying out a process of well-considered steps. In fact, data mining is a step in the larger process of Knowledge Discovery from Databases (KDD) (Han et al., 2011). However, it is also being used as the name of the process as a whole. This process includes cleaning and integration, selection and transformation, (actual) data mining through applying techniques and evaluation and presentation of knowledge gained from the process.

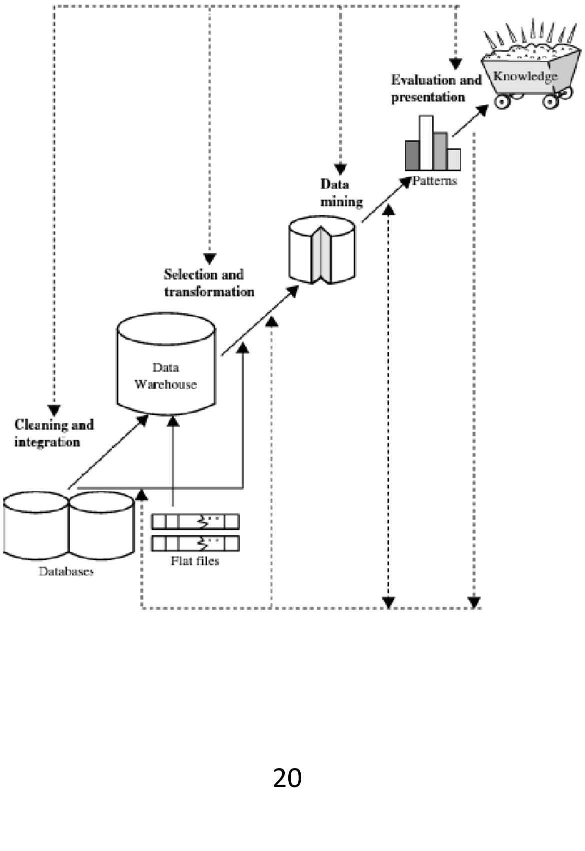


Figure 2.1. Process of data mining (Han et al., 2011)

Based on the process mentioned above, Baesens et al (2015) established another process that is designed for formulating the steps in the fraud analytics. Their “fraud analytics process model” includes seven stages:

a. Identify business problem: Before starting the analysis of data, business problem should be thoroughly identified.
b. Identify data sources: All relevant data sources should be determined for maximum precision of the model. It is stated that when it comes to data, bigger is better.
c. Select the data: All relevant data from data sources should be identified and gathered in a data warehouse.
d. Clean the data: Data should be processed to eliminate any inconsistency such as missing values and duplicates.
e. Transform the data: Data should be reviewed considering the needs of transformation such as binning, aggregation or numerical coding for alphanumeric values.
f. Analyze the data: This is the step where actual model of fraud detection is constructed. Based on the preprocessed data and our goals, a data mining technique is chosen and applied to the data.
g. Interpret, evaluate and deploy the model: In this step, fraud experts interpret and evaluate the model based on real-life conditions and required modifications are made. Following a thorough evaluation, the model can be deployed for fraud detection (Baesens et al., 2015).

2.2.2. Preprocessing in data mining

The five steps mentioned above is defined as preprocessing steps. They are performed to prepare required data for actual analysis. It is generally accepted that 80% of the total efforts in a model preparation is spent on preprocessing. It is very important to perform required preprocessing steps carefully to reach an acceptable model, expressed by the GIGO (garbage in garbage out) principle. In this section, we will summarize basics of preprocessing particularly from Baesens et al. (2015).

Relevant Data Sources

Data can be gathered from many sources, however, mentioning some of the sources may provide insights for fraud detection.

- **Transactional data:** Key attributes of customer transactions form transactional data. It can be used to create trends or can be aggregated to state statistics such as averages, maximums, minimums etc., which are the basic of RFM (recency, frequency and monetary) variables. RFM variables can be used in many ways to understand legitimate customer behavior and detect possible outliers. Some RFM variables may be number of transactions per month, time between transactions, monetary value of transaction vis-à-vis average or maximum etc.
- **Contractual, subscription or account data:** Beside transactional data, institutions gather other more personal data due to the requirements of business or marketing purposes. That data includes sociodemographic information such as age, gender or marital status. For companies, foundation, sector or activity type are example of such data.
- **Data poolers:** Some companies, or government institutions, are specialized in collecting, processing and selling the data regarding individuals and firms. In Turkey, for instance, KKB collects data regarding credit history of individuals, score that data for final users and sell it to the institutions or individuals in line with legal regulations (Baesens et al., 2015).
- **Behavioral information:** Preferences, trends and usage variables form behavioral information. For organizations, turnover or number of employees are examples
- **Unstructured data:** Although it is very difficult to process, data embedded in text documents or multimedia contents can be included in the detection models.
- **Contextual or network information:** In a network setting, context of a particular entity may form a data source which can be used in some of detection models, particularly social network analytics (Baesens et al., 2015).

Exploratory Analysis

After determining data sources, getting an initial insight into the data should be the next step in preprocessing. The distribution of data may be visually examined using pie charts, histograms, scatterplots etc. Some of the unknown relationship between data elements may be discovered in that stage, particularly if the data is flagged as fraudulent/not fraudulent (Baesens et al., 2015).

In data analysis, frequency distribution of the first digits of numbers may provide interesting insights. According to Benford's Law, the frequency of the first digits in many data sets imply that in a typical data set, the count of items will diminish while the first digit of numbers increase. The theory states that 30.1% of the items start with 1, 17.6% start with 2, and only 4.6% starts with 9.

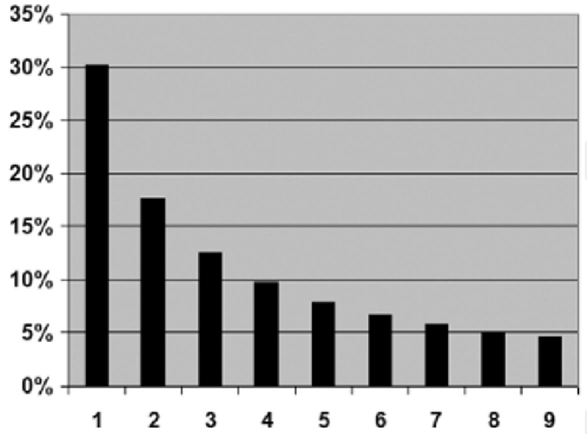


Figure 2.2. First leading digit distribution according to Benford's Law (Kossovsky, 2014)

Benford's Law may be used to detect heavily manipulated data sets, if the data set is normally expected to conform the distribution. It may be a sign of fraudulent or erroneous data in some cases, hence it may be considered as part of the exploratory analysis to spot irregularities.

Beside visual exploratory analysis and Benford's Law, descriptive statistics such as mean, median, mode, percentile values, minimum and maximum values can be examined to reach a "gut feeling" for the data at hand (Baesens et al., 2015).

Handling Missing Values

Missing values are frequent problems in many data sets. They may be the result of incomplete data, privacy issues, errors in data manipulation or inapplicability of the data. The strategies can be replacing (imputing) values, deleting observations with missing values or keeping them if a relationship with target has been discovered (Baesens et al., 2015).

Handling Outlier Values

Extreme observations may occur in the data set. They may be either valid observations, such as the income of a millionaire person on an income data set, or invalid observations, such as height of a person being 4 meters. On the other hand, outliers can be multivariate, i.e. more than one dimension. For instance, having high income when the age is young forms a multivariate outlier, which can be seen on a scatterplot. Beside scatterplots, histograms and box plots can be used to visually identify outlier records. Calculating z scores is another frequently used tool to identify outliers (Baesens et al., 2015).

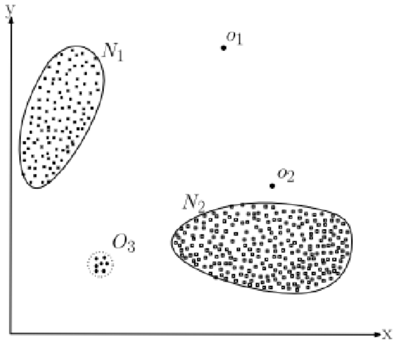


Figure 2.3. Outlier values in a 2-D scatter gram.
(Chandola et al., 2009)

After outliers are detected, invalid outliers may be handled as missing value. For valid outliers, depending on the data mining technique to be used, they may be truncated to acceptable limits through various methods (Baesens et al., 2015).

Standardizing Data

For some data mining techniques, it is required to scale variables in a similar range. For instance, income variable should be standardized to make it a component for logistic regression analysis. Min/max standardization, z-score standardization and decimal scaling methods are alternatives of standardization (Baesens et al., 2015).

Categorization

Categorization helps reducing the number of categorical variables to reasonable levels. For continuous variables, it helps reflecting non-linear effects of variables on linear models. There are various methods to apply categorization (or binning) starting from equal frequency or equal interval binning to use of Chi-square analysis to determine bins (Baesens et al., 2015).

Weights of evidence

Although categorization reduces the number of variables, it means more than one variable for the same category. For instance, categorization may mean 5 categories, hence 5 variables, for age. Weights of evidence is a method aiming at assigning percentages to those 5 variables to merge them. This way, the model become more concise, but at some cost of interpretability (Baesens et al., 2015).

Variable selection

Data mining models start with lots of variables, but generally the few of them have actual explanatory power. To determine explanatory variables, various filters are used to measure correlations between each variable and the target. For different target/variable combinations, several filters are recommended in the literature. For instance, Pearson correlation measure may be used for selecting continuous variables for continuous targets. Filters are very useful for selecting variables, but other considerations such as the correlation between variables, regulatory compliance, Privacy issues and operational attributes such as computation costs should also be taken into consideration (Baesens et al., 2015).

Principal Components Analysis

In a data mining model, many initially defined variables may have correlation with each other. Also known as multicollinearity issue, that phenomenon reduces the stability of the model. If correlated variables can be combined and summarized through principle component analysis (aka primary component analysis), resulting indices would be uncorrelated with each other and increase the explanatory power of the model. In this transformation, interpretability will be impaired due to reduced meaning of new principal component variables (Baesens et al., 2015).

Segmentation

Before initiating model preparation, data may be segmented for various reasons. The first reason may be attributes for one type of records are substantially different from others and, hence, a specific model may be tailored for that segment. Other reasons are interactions between variables and specific strategies for each segment. Segmentation could be very useful, but at the cost of increasing the cost of modelling efforts due to more than one model (Baesens et al., 2015).

2.2.3. Descriptive analytics-unsupervised learning

Descriptive methods aim at “finding unusual anomalous behavior deviating from the average behavior or norm”. That anomalous behavior may also be called as outlier, which is partly explained in the data preprocessing stage. In a fraud detection setting, descriptive methods are used to indicate suspicious transactions. Because they aim at finding anomalous behavior for whatever reason, they can spot irregularities that can’t be indicated by supervised methods, which learn from previous experiences. On the other hand, there are also some challenges experienced by descriptive analytics. Defining “normal” may not be clear-cut in many instances and it changes throughout the time. The attempts of the fraudsters to make their misconducts similar to regular behaviors may lead failure of descriptive analytics and all outliers do not represent fraudulent behaviors. These factors indicate the necessity of extensive follow-up and validation for the unsupervised methods (Baesens et al., 2015).

Graphical Outlier Detection Techniques

Partly mentioned in the above section of handling outliers, outliers can be detected on graphs. While histogram and box plots can be used for one-dimensional outliers, scatterplots can be used to indicate two-dimensional outliers. With the help of rotating the graph, scatterplots may also be used to indicate three-dimensional outliers. To help graphical analysis, OLAP cubes may be used. OLAP cubes have capabilities of rolling up (aggregating), drilling down (getting more dimensions), slicing (bringing details for one group) and dicing (fixing values for dimensions and creating a sub-cube). The insights provided by graphs, OLAP cubes and pivot tables may be very beneficial for exploring fraud patterns. They are also increasingly used in the postprocessing and monitoring stage in data mining. Disadvantages of graphical outlier detection methods are their difficulty for multidimensional analysis and the necessity of active involvement of the end-user (Baesens et al., 2015).

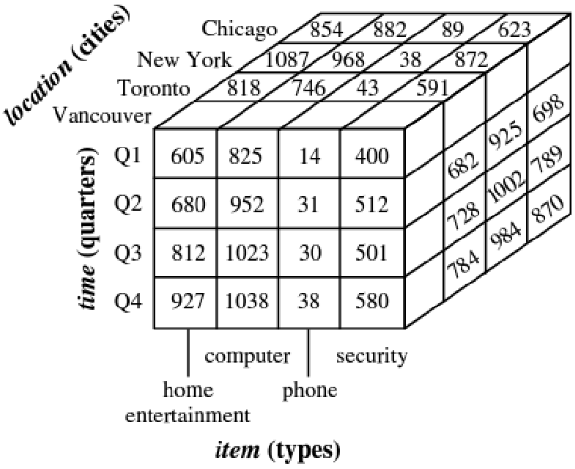


Figure 2.4. 3-D cube representation of sales data
(Han et al., 2011)

Statistical Outlier Detection Procedures

Outliers can be identified statistically through z scores (z scores being more than 3 indicates outlier) or more formal tests such as Grubbs test. Grubbs test also use z scores to calculate one dimensional outlier detection. In multi-dimensional settings, it uses Mahalanobis distance. Grubbs test or other similar tests assume existence of normal or other distributions in the data, which is not always true.

Break Point Analysis

Initially proposed by Bolton & Hand (2001), this analysis is used to detect abrupt changes in an account (it may be a bank account, telecommunication account or credit card). It defines a fixed time window to determine the profile of the account and compares averages of this part with a subsequent smaller time window through t statistics. Observations can be ranked based on the calculated t statistics (Baesens et al., 2015).

Peer group Analysis

This method is also proposed by Bolton & Hand (2001). A peer group may be defined as "a group of accounts that behave similarly to the target account". For instance, middle income wage earners who spend a determined average in their credit cards can form a peer group for credit card expenditures. The exact accounts that form the peer group could be determined by domain knowledge or by statistical analysis. The number of peers should not be too small or too large for meaningful results. The behavior of the individual account is compared with its peers through calculating t statistics or Mahalanobis distance, depending on the number of dimensions (Baesens et al., 2015; Bolton & Hand, 2001).

After determining peer group, the behavior of the account is compared with the overall behavior of the accounts in the peer group. In this way, seasonal variations that can be inferred from peer group could be grasped, which is not possible in break-point analysis. On the other hand, both peer group analysis and break point analysis show local outliers instead of global outliers, which means the behavior of account is considered anomalous because it is found anomalous within the account or peer context, rather than the universe of all accounts.

Association Rule Analysis

Association rules aim at analyzing frequently occurring relationships between items. In its origin, association rules were used to find items frequently purchased together. In fraud detection, it can be used to find frequent item sets and fraud rings in insurance claims. Data scientist firstly determine a threshold of *support* level, which is the determined percentage of "frequent item sets" in the total records. This ratio may be 10%. Aftermath, they determine a *confidence* level which may be defined as "the conditional probability of rule consequent, given the rule antecedent". The determined frequent item sets may expose some irregularities, as in the case of fraud rings (Baesens et al., 2015).

Clustering

Clustering aims at splitting a set of observations into segments. Its objective is to maximize homogeneity within each segment (cohesiveness), while minimize homogeneity between segments (separated). In fraud detection settings, clustering can be used in many fields such as credit card transactions or insurance claims. Many types of data including account data, unstructured data, behavioral data, or RFM (recency, frequency and monetary) variables may be used in fraud detection settings.

In the calculated clusters, fraud should be considered in particularly sparse and small clusters. Clustering techniques require correlated inputs to be removed as much as possible. It can be categorized as hierarchical or non-hierarchical clustering at the main level (Baesens et al., 2015).

In the calculation of clusters, distance metrics provide the basic for quantifying similarity. Various distance metrics were accepted for different type of variables. For continuous variables, Euclidian distance or Manhattan (City block) distance may be used. In order to reach meaningful conclusions, variables should be standardized to make their ranges similar.

Hiearchical clustering, one of the main groups of clustering methods, aims at hierarchically forming clusters. It starts from either individual observations (agglomerative clustering) or set of all observations (divisive clustering) to reach necessary number of clusters. The number of clusters is not set before starting analysis and this is an advantage of the method. However, it is not appropriate for large data sets and interpretation of the resulting clusters require subjective expert analysis (Baesens et al., 2015).

Partitioning clustering does not iterate hierarchically. For these methods, an initial number of clusters are determined before starting the analysis. Later, the software uses iteration to reach optimal cluster centroids. One of the most used non-hierarchical clustering methods is k-means clustering. In k-means clustering, mean of the observations is chosen as the centroid of the clustering. Other versions of the method use medians, which is less sensitive to outliers, and named as K-medoid clustering. For categorical variables, k-mode clustering uses modes for determining centroids (Han et al., 2011). The working of k-means clustering family is as follows:

- Identify k observations as seeds (initial cluster centroids)
- Assign each observation to the closest cluster (using Euclidian distance)
- After assigning each observation, recalculate the centroids position
- Repeat until the centroids no longer change or adequate number of iterations reached (Baesens et al., 2015).

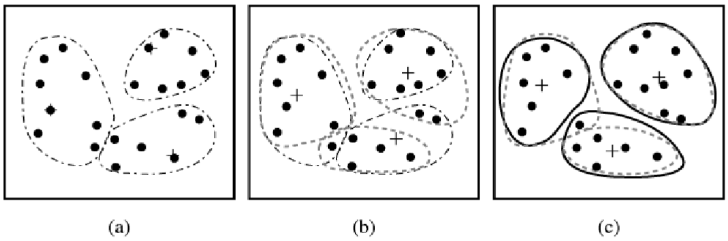


Figure 2.5. Iteration steps in k-means clustering
+ Denotes centroids (Han et al., 2011)

Hierarchical and partitioning clustering assumes spherical-shape clusters. For clustering based on arbitrary shape, density based clustering algorithms (such as DBSCAN) were produced (Baesens et al., 2015). Beside these three groups, grid and model clustering were discovered for different purposes (Ahmed et al., 2016).

Self-Organizing Maps

A relatively new method of unsupervised learning, SOM method enables visualizing and clustering high dimensional data on a low dimensional grid of neurons. Neurons in the output layer are ordered in a two-dimensional rectangular or hexagonal grid. They are handy for visualizing high-dimensional data. The downside of SOM is the requirement of expert knowledge to interpret optimal size of the SOM and to compare different SOM solutions (Baesens et al., 2015).

Evaluating Clustering Applications

Different applications of clustering algorithms may be evaluated statistically or logically. In statistical evaluation, sum of squared errors may be used to aggregate the distance between cluster centroids and observations. In terms of interpretation, distributions of observations regarding a variable in a cluster can be compared to the distribution in the overall data (Baesens et al., 2015).

2.2.4. Predictive analytics-supervised learning

Predictive analytics aims at building models to predict a target. In a broad sense, they can be divided into regression and classification. Regression is used to predict a continuous interval, such as 0 and 1 or such as 0 and infinity. Fraud amount can be a target for regression. On the other hand, classification uses categories as target measures. Thus, they can take a limited set of predefined values. For instance, fraud vs. no fraud can be a target variable for binary classification. In multiclass classification, more than two targets can be chosen such as severe fraud, medium fraud and no fraud. Different methods can be used together to solve complex problems such as fraud (Baesens et al., 2015).

The main challenge of predictive analytics in fraud detection field is inability to certainly flag observations as fraud or no-fraud. Flagging fraudulent transactions may be difficult due to the determining a higher threshold for examined observations and difficulty of examination. Because it is almost impossible to flag all fraud cases, models should take this into consideration (Baesens et al., 2015).

Linear Regression

To predict continuous target variables, linear regression is generally the first choice. It can be used to predict fraudulent amount in a fraud detection setting. The generalized formula of the regression is $Y = \beta_{0+}\beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_N x_N$. In this formula, Y represents the target variable and x_1, … , x_n represents explanatory variables. The simplest form of linear regression is ordinary least squares (OLS) regression, which is easy to understand. More sophisticated versions of logistic regression, such as ridge regression or time series models aim at reducing the linearity assumption.

Logistic Regression

In order to predict binary variables (0-1), a modified version of linear regression should be used due to the attributes of OLS. Using bounding function modification, linear regression formula is converted to logistic regression, which provides targets between 0 and 1. In that sense, logistic regression provides a probability for the target. Probit or cloglog transformations are alternatives to logistic (logit) transformation of target. Once the parameters are estimated, logistic regression can be easily converted to a point-based fraud score calculation method which is very handy for straightforward interpretation and prioritization of observations at hand (Baesens et al., 2015).

Variables of linear and logistic regression are selected based on statistical tests which calculates whether coefficient of each variable is different from zero. However, beside passing statistical tests, several more criteria should be met for variables. First of all, the sign of the coefficients should be in line with the expectation of the expert to prove the interpretability of the variables. Furthermore, operational efficiency should be provided particularly for real-time detection solutions. Finally, legal issues should be taken into consideration regarding the collection of variables.

Decision Trees

Decision tree algorithms come up with a flowchart-like tree structure to represent patterns in the analyzed observations. The top node of the tree is known as a root node. Each node identifies a testing condition. The outcome of the test leads a branch which forms an internal node. Terminal nodes of the tree, also known as leave nodes, assign labels. Most popular decision tree algorithms are C4.5, CART and CHAID. In a decision tree, following two decisions should be made:

- Splitting decision: Which variable should be split at what value
- Stopping decision: When to stop adding nodes (Baesens et al., 2015)

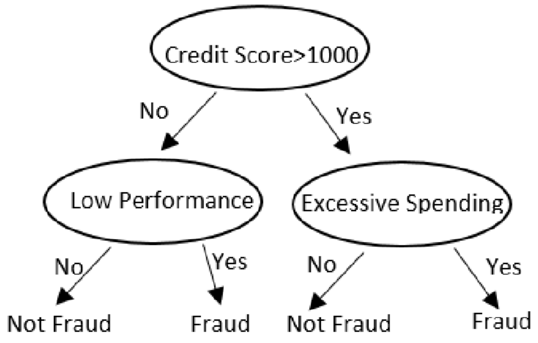


Figure 2.6. A part of decision tree predicting employee fraud

In splitting decision trees, the algorithm try to minimize impurity of labels in the child nodes. Various measures of impurity are being used such as Entropy, Gini or Chi-square. In stopping decision, the tree's performance, which is trained through training sample, is tested on separate part of data, which is named as validation sample. Until some point, misclassification reduces on both training and validation samples. This point is defined as the stopping decision, beyond which decision tree *overfits* training sample and increase misclassification for validation sample.

Neural Networks

Neural networks may be defined as generalization of existing statistical models. As can be seen in Figure X, they include a hidden layer between input and output layers, which make their interpretation difficult. As a black box method, they make the connection between inputs and outputs "mathematically complex and nontransparent" way (Baesens et al., 2015).

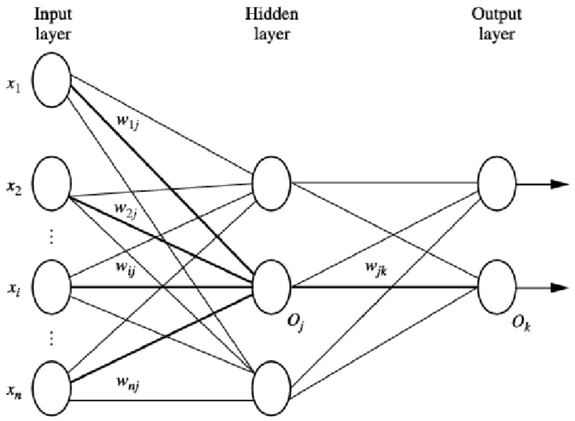


Figure 2.7. Multilayer feed-forward neural network
(Han et al., 2011)

Although there are some methods which can partially expose inner working of a neural network, they remained less interpretable methods for fraud detection. The advantages of neural networks are their tolerance to noisy data, their ability to classify without expert or domain knowledge and their aptitude to many real-world problems (Baesens et al., 2015).

Support Vector Machine

Support Vector Machine (SVM) is another advanced classification method working by transforming a linear issue into a higher dimensional feature space. This property provides the method to reach linear solution for complex non-linear problems, such as fraud detection, without computational complexity (West & Bhattacharya, 2016).

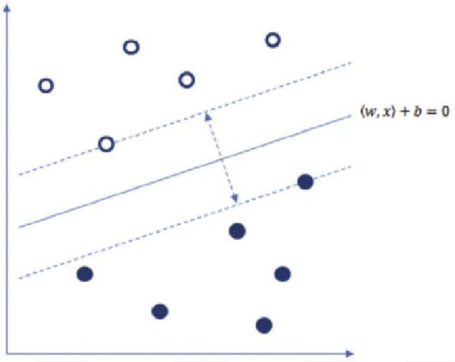


Figure 2.8. A two-dimensional support vector machine representation (West & Bhattacharya, 2016)

Ensemble Methods

Ensemble methods were created with the assumption of "multiple models can cover different parts of the data input space and as such complement each other's deficiencies". Most popular implementations of ensemble methods are Bagging, Boosting and Random Forests. Bagging (bootstrap aggregating) works by taking B bootstraps from the sample and building a classifier for each bootstrap. The final classification is achieved by the votes of all bootstrap classifications. Boosting works by weighting the data according to misclassified observations. Random forests work by creating subset of the data, like in Bagging, and performing classification using a subset of attributes (Baesens et al., 2015).

As a popular ensemble method, random forests are devised to learn from subsets of data, thus to prevent overfitting, by considering certain number of fully-grown CART decision trees. They are used for both classification and regression. They train certain number of trees and give results based on the mode (classification) or average (regression) of the trained trees. Their popularity is especially evident in categorical variables vis-a-vis continuous variables (Bhalla, 2014).

Random Forest algorithm works in the following order:

a) Randomly selecting records: Each tree is trained using a 63.2% sample of originally available training data. This random sample is redrawn with replacement from training data for each tree.

b) Randomly selecting variables: Out of all predictor variables, some of them are selected randomly. For classification, this value is square root of the number of all variables. For regression, it is 1/3 of the number of all variables. This parameter can be changed in formula.

c) Leftover data (36.8%) for each tree is named as out-of-bag (OOB) data and misclassification rate on OOB is calculated for each tree. Aggregate error from all trees is used to indicate OOB error rate of the forest.

d) Each tree gives a classification (classification) or value (regression) for OOB observations. For classification, we can say that each tree votes for OOB observations and the ratio of votes to total votes (number of trees) for each OOB observation will determine the score given by this random forest for that class. For regression, average of values given by trees will produce model results (Bhalla, 2014).

Random forest can deal with high-order interactions, few observations but many variables problems and correlated variables. Beside prediction, assessing variable importance is one of their advantages (Strobl & Zeileis, 2008).

RIDIT and PRIDIT

RIDIT is a type of data standardization method used for categorical rank-ordered variables. Originally introduced by Bross (1958), it incorporates ranked nature of variables and observed response probabilities. Through this method, a categorical variable containing rank could be standardized within the range of -1 and 1 (Brockett et al., 2002). The equation of the RIDIT score is given below:

Equation 2.1. Computation of RIDIT Scores

$$B_{ti} = \sum_{j<i} \hat{p}_{tj} - \sum_{j>i} \hat{p}_{tj} \quad i = 1, 2, \ldots, k_t.$$

Source: Brockett et al. (2002)

In this equation, t denotes the ranked categorical variable, k_t number of ranks in variable t, $\hat{p}_t$ =($\hat{p}_{t1}$,.., $\hat{p}_{tkt}$) observed proportions of responses for variable t in the whole data set. Response categories should be ordered in the same manner. In fraud detection setting, higher categorical response should mean less fraud suspicion. The result of this equation would be in the range of -1 and 1.

Brockett et al. (2002) stated that the RIDIT method have some advantages vis-à-vis other methods handling rank-ordered categorical variables. First of all, it doesn't treat categorical values like they are interval values, as in the most usages of natural integer scoring in many research fields. Furthermore, it doesn't require assumptions of other methods such as the equality of space between categories and conformity of distribution of the input data. Moreover, it reflects the relative abnormality of the response. Besides, this scoring method is "monotonically increasing" which corresponds the ranked nature of responses. Finally, for each variable, score would be centered and total of scores would be zero (Brockett et al., 2002).

Brocket et al. (2002) suggest that principal components analysis on RIDITs (PRIDIT) may serve as an objective suspiciousness scoring for fraud detection. Their proposal claims that principal component arisen from PCA would represent the variables best and thus can be used as a classification function. They assert that such a method would overcome several disadvantages of other predictive methods regarding its assumptions on requirement of training samples, its ability to distinguish information value of variables, and its power to accurately classify rank-ordered categorical variables (Brockett et al., 2002).

Evaluating Predictive Models

Predictive models are evaluated by splitting data at hand to two parts: training data and test data. They should be completely separated for a successful test. Beside these two sets, validation data sets should also be split for decision trees or neural networks. Validation data set is used to determine stopping decision. All of these data sets are expected to be formed through stratified sampling on target variable. For small data sets, there are other options for determining those splits (Baesens et al., 2015).

After splitting data sets, a performance measure is determined. A receiver operating characteristics (ROC) curve is one of these measures. Based on the concept of confusion matrix, which compares predicted status with actual status, area under ROC curve (AUC) is used to compare different predictive models. In the following figure, M1 model seems producing better predictions (in AUC terms) than M2.

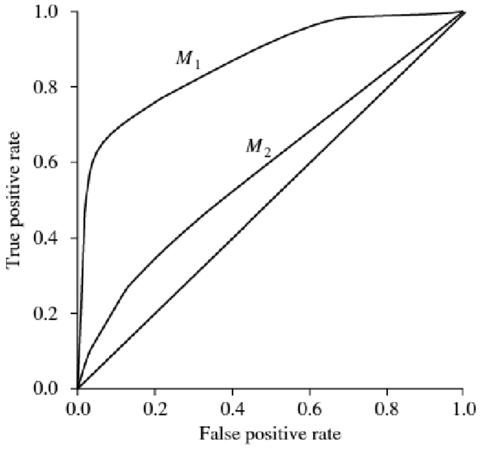


Figure 2.9. Sample ROC curve comparing two models
(Han et al., 2011)

Beside AUC, there are various measures for model performance. The lift curve measures percentage of fraudsters per decile of observations. Cumulative accuracy profile (CAP) curve is another representation of the lift curve. Based on the CAP, Accuracy Ratio (AR) may be calculated for the model at hand to indicate its closeness to a perfect model. Accuracy ratio is also known as Gini coefficient. As stated before, statistical measures of model performance should be considered with other factors, such as comprehensibility, justifiability to previous business knowledge, and operational efficiency (Baesens et al., 2015).

Developing models for skewed data sets

In fraud detection model, one of the most significant challenges of fraud datasets is skewed data sets. A summary made by Baesens (2015) suggests that actual fraudulent observations may be as less as $5*10^{-5}$. The inadequate ratio of flagged observations may lead performance measurements to indicate erroneous results. To overcome this problem, either number of fraudulent observations or their weights may be increased. Several methods such as undersampling, oversampling, SMOTE and cost-sensitive learning are proposed for correcting the skewed data sets to a certain extent (Baesens et al., 2015).

Oversampling works by replicating the fraudulent observations. On the other hand, undersampling works by reducing the number of non-fraudulent observations. Inactive accounts or low-value transactions can be first candidates for reducing legitimate observations. Synthetic Minority Oversampling Method (SMOTE) synthetically add flagged data with the k-nearest neighbor calculation. Cost-sensitive learning changes the assumption of equal costs for the cases in confusion matrix and increase the cost of false negatives to reach a better prediction in a fraud detection setting (Baesens et al., 2015).

2.2.5. Other techniques

Social Network Analysis

Social networks have been a phenomenon of the last decade. Beside being the most prominent form of social activity, social networks provide a rich data set for data analysts, as far as it is allowed. In fact, social network analysis is not confined to popular social networks over internet and can be used for entities in enterprise data. For instance, having massive transactional databases, telecommunication companies can easily spot the strength of relationship between people based on the frequency and/or duration of calls (Baesens et al., 2015).

In fraud context, social networks may be very useful because of the requirement of collaboration for committing fraud. In the lack of required evidence for fraud, analyzing the network of people involved in the cases may provide important contributions to examine fraud. In insurance claims, fraudsters may employ same set of witnesses, claimers, claimees or vehicles. Examination of tax fraud cases can also benefit from social network analysis. In employee fraud context, some fraud types, which require collaboration between employees and other actors, social network analysis may be used (Baesens et al., 2015).

Social network analysis can be represented on unipartite, bipartite or multipartite graphs. Unipartite graphs include one type of entity. The relationship between people in a social network application is represented on a unipartite graph. Bipartite graphs include two types of entities, for instance, in credit card fraud, cardholders and merchandisers can be linked on a bipartite graph. Multipartite graphs are quite complex structures for analyzing and they are not used much in fraud detection context (Baesens et al., 2015).

Genetic algorithms and programming

Using the concept of evolution, genetic algorithms iteratively improve solutions for the problem. It starts by creating a starting generation randomly, then continues with reproducing each population using various techniques. Finally, it chooses survivors based on their fitness. To measure the fitness of the "children", percentage of correctly classified samples are used. The algorithm would finish when a required fitness has been reached or a definite number of iterations were made (West & Bhattacharya, 2016).

Text Mining

Text mining is performed on unstructured text data to reach a classification. In the first step, various preprocessing activities are carried out to reach a more quantitative representation of the text. Such preprocessing activities may include filtering some words, eliminating prefixes and suffixes from stemming and measuring the frequency of words (West & Bhattacharya, 2016).

Process Mining

Process mining aims at constructing models that can represent the behavior of a system. In its typical usage, the first step involves the preparation and inspection of logs together with cleaning extraneous noise. A model is built upon the processed logs. In analysis stage, model is observed for expected outcomes of the system. Finally, process miner is applied to various samples and determines their conformity to typical system behavior (West & Bhattacharya, 2016).

2.2.6. Challenges and Success Factors of Fraud Detection

As Baesens (2015) stated, there are several characteristics that need to be satisfied by a successful data mining model for fraud detection. Provision of these requirements is quite challenging, yet compulsory.

- **Statistical accuracy** can be defined as explanatory power and correctness of the produced model. Accuracy may be proven by criteria such as hit rates, lift curves or AUC. Furthermore, the model should produce generalizable results, thus avoiding overfitting to the data at hand.
- **Interpretability** may be a requirement for a model if it is required to understand why a case is flagged as fraudulent. This is especially important for models in a validation stage. Some of the data mining techniques provide white-box models, which enables revealing the reasons of flagging, while others provide black-box models, which are mathematical and incomprehensible for human being.
- **Operational Efficiency** may be a requirement if the model should evaluate a case in a short time period. In some cases, it may also refer to the efficiency of efforts in

the preprocessing, evaluation, backtesting and reestimating stages (Baesens et al., 2015).

- **Economical Cost** should be taken into consideration while developing a detection model. Beside the costs of model development, indirect costs such as human and software costs should be taken into account in a thorough cost-benefit analysis before the model development.
- **Compliance** may be required if there exists some legislation relevant for the model, or some privacy rights may prevent collecting required data (Baesens et al., 2015).

Beside those success factors, following considerations should be analyzed in a fraud detection model.

- Fraudsters try to be innovative to beat existing detection methods, hence, fraud detection models should have the ability of rapidly adapting themselves to new cases (Baesens et al., 2015).
- While increasing the accuracy of the model, it is also important to keep false positive rate to be as low as possible to prevent harassing good customers.
- Analytical models generally have a challenge to cope with class imbalance/data skewedness problem, for instance the flagged cases in credit card fraud are generally less than 0.5% (Baesens et al., 2015). Due to this imbalance, misclassification costs should be carefully considered in the analysis (West & Bhattacharya, 2016).

With respect to anomaly detection techniques, Ahmed et al (2016) adds following issues in data mining:

- The appropriate data analytics technique should be evaluated thoroughly, because attributes of normal and abnormal classes change according to the data.
- Data itself contains noise and that noise may be revealed as an outlier in the data set, which hinder discovering the true anomaly.
- Normal behavior for one account may legitimately change throughout the time (Ahmed et al., 2016).

2.2.7. Deciding to Establish a Fraud Detection System

Despite the first sight appeal of fraud detection through data analytics, establishing a fraud detection system should be considered carefully by taking into account the costs and benefits of such a system.

First of all, the actual utility of fraud detection to the firm should be analyzed. In this analysis, indirect impacts are needed to be included beside apparent direct impacts. Baesens et al. (2015) states that utility of examining a fraudulent record is higher than examining a non-fraudulent record. But the latter has also utility due to "scare-off" effect of raising fear for perpetrators regarding detection. The other factors regarding the measurement of utility of fraud detection are; (a) Amount of fraud, (b) all involved costs in the fraud detection including legal costs, investigation costs, cost of detection infrastructure (c) penalties or fines as a result of fraud, (d) noneconomic values such as benefit to the society and to the firm.

As Vona (2008) stated, the detection of fraud in regular audit framework has important limitations. On the other hand, considering the required domain knowledge and "data science" expertise, the cost of establishing and operating a fraud detection system is not trivial. Furthermore, such a system should also include establishing a framework to facilitate detection and to handle fraud case management (Baesens et al., 2015). In determining the scope of the project, it may be crucial to apply Pareto principle of diminishing returns and start from most feasible fields.

Baesens et al. (2015) provides a framework for determining the cases to be examined. The authors consider the fraud probability for each item and multiplies this probability with the amount of transaction to find an expected loss, which can be used to help deciding which case should be investigated. Hence, a suspicious transaction with higher value but medium fraud score may be more likely to get priority than a transaction with high fraud score but low value.

2.3. Literature review

In this section, we will review the relevant literature regarding fraud detection through data analysis and data mining. We will explore most relevant surveys of fraud detection literature, and emphasize the particularly important articles for credit and debit card fraud.

2.3.1. Surveys of Data-Driven Fraud Detection

Due to the frequency of data-driven fraud detection literature, we will examine outstanding surveys of fraud detection studies. Some of these surveys are confined to fraud field (such as credit card fraud) whereas some others are based on detection methods (such as clustering methods). Few studies provide inter-disciplined research involving different data mining methods.

West & Bhattacharyya (2016) provides an in-depth survey of fraud detection research over fifty relevant articles from the period 2004-2014. They give a comparison of articles with regard to their field of study and data mining methods. The authors preferred the term of "computational intelligence" rather than data mining. They suggest that the data mining literature is having more diverse techniques in the last decade compared to the previous decade (1994-2004). The authors indicate neural network techniques and logistic regression as the most established history regarding fraud detection and they can be used as a comparison tool when using other data mining methods. There is also a comparison between data mining methods considering their strengths and limitations. The issue of misclassification costs, which is an important problem of fraud detection research was also discussed regarding each detection methods. The authors provide a concise summary for data mining techniques for fraud detection. They mention the usage of hybrid detection methods, which are composed of different techniques and fuzzy logic. They suggest that the usage of advanced techniques like genetic algorithms remained limited. Finally, they indicate some challenges of fraud detection researches, including lack of data due to privacy considerations and disproportionate misclassification costs along with typical classification problems such as feature selection and tuning of the model. They finally propose establishing a general framework for fraud detection that can be used in all fraud detection models (West & Bhattacharya, 2016).

Sabau (2012) reviews 27 articles on various fraud detection which use clustering techniques. His study includes diverse financial fraud areas ranging from credit card fraud to insurance fraud to corporate fraud. He states that several studies use standalone clustering techniques, whereas some others are complex data mining implementations which includes more than one technique and or stage. In some studies, clustering visualization techniques were also used. In standalone clustering studies, k-means algorithm is being used with Euclidian distance as dissimilarity metric. In an interesting research, Little (2008) uses Benford Law to find unusual group of healthcare payments and consequently applies clustering technique to find fraudulent payments in this unusual group. Another strand of research used clustering to label fraudulent and not fraudulent data and used these labels to run another classifier like neural networks and decision trees (Little et al., 2008; Sabau, 2012).

In some researches, clustering techniques are used to group already labelled entries by other classifiers. The goal of that usage is to define a taxonomy of the already identified fraudulent entries in order to determine measures to prevent them. Some researches use hierarchical clustering techniques (such as BIRCH) together with k-means algorithm to benefit from their strong dimensions. Some of the researches used density based (such as DBSCAN) algorithm together with resolution based (such as k-means) algorithms. Visualization using variable binned scatter plots is another interesting usage of clustering technique for fraud detection (Sabau, 2012).

Chandola et al (2009) provides a comprehensive overview of anomaly detection research, which is a common method group used in many fraud detection algorithms. They included the anomaly detection research in the fields of credit card, mobile phone, insurance and insider trading frauds. They make a distinction between point anomalies and contextual anomalies. While point anomalies are defined as anomalous records with respect to the rest of the data, contextual anomalies are anomalous in a specific context. In terms of credit card fraud detection, a credit card payment which is substantially more than any payment of that customer would mean a point anomaly. On the other hand, existence of a high payment amount in business days can be contextually anomalous if the user’s behavior shows such payments only on holiday periods. Contextual anomaly techniques require existence of contextual attributes like spatial, graphs, sequential and profile. The authors suggest that contextual anomalies could be solved by either reducing them to point anomaly or, a more difficult solution, modeling the anomalous structure in the data (Chandola et al., 2009).

Ahmed, Naser, & Islam (2016) provides another survey on clustering based anomaly detection research in financial domain. They provide three assumptions of clustering techniques that was utilized, to a certain extent, in the relevant research. The study differentiates clustering techniques as partitioning, hierarchical and other groups. Their survey indicated that the research which use partition based clustering methods utilized various versions of k-means algorithm. The application fields vary from telecommunication to money laundering to insider trading. Generally, the small clusters are identified as fraud-prone and further investigated by experts. The datasets are not available mostly due to privacy issues. In hierarchical clustering techniques, three articles were reported which aim at finding non-compliant financial reports and information fabrication in foreign trade transactions. In miscellanous clustering techniques, DBSCAN, latent clustering or resolution based clustering methods were used on purchasing orders, online shopping, policy holder or synthetic data. The authors also discuss the usage of synthetic dataset in fraud detection domain, an apparent need due to the privacy issues largely encountered in real fraud data sets.

Another important research direction focuses on signatures to detect fraud particularly in temporal data (Edge & Falcone Sampaio, 2009). This direction asserts that supervised data mining techniques like neural networks or Bayesian learning require "extensive training using labelled data sets for formulation of evaluative models against which to assess newly arriving transactional instances." For this reason, new fraud cases may not have detected (due to time consuming and costly business operation) and detection could only be achieved after the transaction have completed. As an alternative, various institutions, including software companies, are trying to produce proactive fraud detection methods which can trigger preventive response before transaction completed. That requires "maintenance of a statistical representation of user behavior against which to evaluate new system transactions and their likelihood of representing a fraudulent transaction". Following a notification of transaction, that statistical representation, which is called "signature", may be recalculated and be compared to previously held values for sizeable deviations from normal behavior. In fact, the context of signature resembles "activity monitoring" approach in the earlier fraud detection literature (Fawcett & Provost, 1999).

Signature based detection methods have evolved from general thresholds (which resemble rule-based fraud detection) to segment based thresholds to customer based profiles. It is emphasized that customer based profiling is a necessity to help fraud detection. The running time of signature algorithms are also discussed. In real-time detection, event-driven signatures should be calculated in each account movement. However, this may be computationally difficult. Time-driven signatures, on the other hand, are calculated on certain time periods, like hours, days or weeks (Edge & Falcone Sampaio, 2009). Various version of signature based methods are used in several articles (Cahill et al., 2002; Cortes & Pregibon, 2001; Ferreira et al., 2006; Xing & Girolami, 2007).

In Turkish academia, several thesis were prepared on issues such as fraud risk (Varıcı, 2011), fraud types and fraud prevention (Şengür, 2010), procurement and payment cycle frauds (Çetin, 2013) and profile of fraudsters (Gündüz, 2014).

Some papers primarily focus on innovative techniques which can be considered useful in fraud detection. Graph based anomaly detection and visualization techniques are one of the most interesting direction in that regard (Akoglu et al., 2015; Argyriou et al., 2013; Eifrem, 2016). Determining fraud in networks through analysis of connections is another future direction in data mining (Martens et al., 2013; Van Vlasselaer et al., 2016).

Class imbalance problem, which should be considered when creating a fraud model with little existing "positive" labels, is also analyzed in several papers (Lazarević et al., 2004; Longadge et al., 2013; Mosley, 2013; Sáez et al., 2016; Satyasree & Murthy, 2013).

Another important topic which we will encounter in our study is creation of new features in the data. There are several articles which deal the problem of so called "feature construction". (Bahnsen et al., 2016; Lee & Stolfo, 2000; Motoda & Liu, 2002; Wen et al., 2014).

2.3.2. Important Researches in the Plastic Card Fraud Detection

Debit card fraud has become the research subject rarely. On the other hand, many attributes of the problem resemble to credit card fraud detection. Credit card fraud detection became a popular problem among researchers since 1990's. Google scholar search for the exact phrase of "credit card fraud detection" returns as many as 4.000 results since 2000. In this section, we will review some of the most prominent articles and discuss the issues frequently mentioned in the literature regarding debit and credit card fraud detection.

The interest toward credit card fraud detection seems to have started in 1990's. We reviewed two articles in this period. Chan et al. (1999) discusses combining multiple classifiers together by meta-learning. Each classifier (CART, C4.5, Ripper and Bayes) is run in different subsets of data, which is gathered from two commercial banks. The authors determined around 30 variables regarding transaction. The classifiers consider the variable cost of each transaction through an implementation of Adaboost algorithm. Furthermore, the cost of false positives is included in the analysis through a computed "overhead" for fraud detection. Hence, suspicious transactions below that threshold should not be examined. Results are discussed for various distribution of target in training dataset. They reach the conclusion that combining multiple classifiers contribute the efficiency of fraud detection model (Chan et al., 1999).

Dorronsoro et al. (1997) discusses an actively used fraud detection model in a credit card intermediary company. In terms of data, intermediary company do not have long-term account history of credit card owners. With the available short-term account behavior, they applied a non-linear version of Fischer discriminant analysis together with Artificial Neural Networks. Before application of classifier, segmentation was implemented to credit card transaction data to reduce imbalance of fraudulent transactions from less than %0.01 to around %0.6. Although some of the fraud cases were excluded from analysis due to this process, it generally enhances the total detection capability. Class overlapping, which means fraudulent cases may be labelled as non-fraud, is mentioned to be considered due to its detrimental impacts on many classifiers, such as neural networks. They conclude that the method suggested have demonstrated the capability of commercial implementation (Dorronsoro et al., 1997).

Quah & Srinagesh (2008) proposes Self-Organizing Maps (SOM) for classifying and clustering input data and filtering for further analysis in a credit card fraud detection system. In performing SOM analysis, they apply standard Euclidian distances and a proposed gravity function that is used to weight variables against the whole variables. The variables are classified as customer-related, account-related and transaction-related vectors. They suggest that categorical variables may be converted to numerical variables using their frequency of occurrences in the dataset. Later, those variables may be normalized to fall in a specific range. The authors also point out abnormal transactions that may indicate the onset of a fraudulent spending pattern. They also discuss actual implementation of their proposed algorithm in a financial institution's IT system. Finally, it is suggested to further develop their approach for a running detection system (Quah & Sriganesh, 2008).

Juszczak et. al (2008) compares various one-class classification algorithms for fraud detection. They suggest that unsupervised one class classification, as opposed to supervised two-class classifiers, has the ability to detect abnormal changes in customer behavior without known cases. To measure the performance of their approaches, they suggest using a *performance curve* which plot time of the detection and ratio of total costs incurred. They compare the performance of two-class classifiers with one-class classifiers and concluded that although two-class classifiers are better at the beginning, their performance deteriorates due to the population drift associated with changing account behavior. In their experimental setup, they made a distance-based representation of ATMs (Whitrow et al., 2009).

Dal Pozzolo et. al (2014) performed a thorough comparison between different algorithms (random forest, support vector machines and neural networks), different sampling methods (undersampling, SMOTE, EasyEnsemble), different model update frequency (One time, 15 days, weekly and daily), number of model in the ensemble and incremental approach used in the model update procedures. They have used classical transaction variables together with aggregated variables computed within 3 months before transaction. As performance measure, they suggest PrecisionRank, which measure the ranks of positive class among top n records. The authors come up with the results that sampling procedures are important for a successful learning, updating models frequently lead improving performance and random forest models beat neural networks and support vector machines (Dal Pozzolo et al., 2015).

Halvaiie & Akbari uses artificial immune system (AIS) algorithm for detection. Inspired by immune system of people, AIS have applicable features for fraud detection, including the assumption of unbalanced ratio of positives and requirement of adapting to new cases. They propose a new cost function for the performance measurement of the detection model which subtract the TP amount from FN amount and add multiplication of FP records with a determined examination overhead per record. Considering the computational workload of the process, they opt out for a distributed data-processing in Hadoop platform. Their results suggest that a modified version of AIS can be used in credit card fraud detection (Halvaiee & Akbari, 2014)

Ekrem Duman has written several articles together with other co-authors on credit card fraud detection. Duman & Özçelik (2011) proposes using genetic algorithms combined with scatter search for the problem, due to the incapability of classical data mining algorithms for variable cost-sensitive problems such as credit card fraud. Rather than building a new model from scratch, they have taken an active fraud detection model and successfully used GA and SS algorithms for determining weights of the parameters and final score threshold. They also found that the region of card spending considering the user's previous spending regions is quite important for the detection problem (Duman & Ozcelik, 2011).

In Sahin et al. (2013), the authors propose a cost-sensitive decision tree, which uses a variable cost measure to split tree at each non-terminal node. They conclude that the cost sensitive decision tree was able to reach better performance not only with regard to computed cost, but also regarding more general performance measures of accuracy and area under curve. In their approach, performance measure is determined as Saved Loss Rate (SLR), which is the ratio of potentially saved loss to the total loss. They conclude that cost sensitive decision trees perform better than non-cost sensitive algorithms and algorithms assuming a fixed misclassification cost (Sahin et al., 2013).

In Mahmoudi & Duman (2014), a modified version of Fischer discriminant analysis is proposed for the problem, which prioritize (bias) towards more important items. Thus, the aim of the research is transformed to profit maximization. They found that modified Fischer discriminants are superior over decision trees, neural networks, naïve Bayes and normal Fischer discriminant (Mahmoudi & Duman, 2015).

Bhattacharrya et. al (2011) discuss the potential of aggregating transaction variables (through counts and total amounts) for improved detection of fraud. In Bhattacharrya et. al (2011), logistic regression, support vector machines and random forest is compared as classifier. Subsampling was thoroughly applied to train a model from around 50 million records. They include many performance metrics including accuracy, AUC, sensitivity, specificity, precision, harmonic mean of precision and recall. They concluded that random forest performs better than logistic regression and SVM, but the performance is more visible in models which are trained in balanced datasets (they used 15% fraud rate in the most balanced dataset). Employing same dataset, Jha et al. (2012) uses logistic regression as classifier and presents the causality and coefficients of aggregated variables. They suggest that the higher aggregated variables for a shorter time period, the higher the risk of fraud. For longer time period, such as 3 months, aggregated variables generally mean less suspicion for the transaction. They also propose that incorporating time stamps of the operations into the analysis may enhance predictive power of the models (Bhattacharyya et al., 2011; Jha et al., 2012).

Panigrahi et al. (2009) propose a fraud detection system with 4 stages. Their model starts with a rule based filter with some general or customer-specific rules. They indicate address mismatch and outlier detection rules as examples. The fraud probability resulted from rules are combined through Dempster-Shafer Adder (DSA) which is used in fusion of different sensor information. Later, the customer information is summarized in the transaction history database which can compare fraudulent transactions with legitimate transaction. Finally, probabilities reached from transaction history database is used to reach a final opinion through Bayesian learning. Using simulated credit card data, they suggest that their proposed model have the merit of reducing false positive rate in the problem at hand (Panigrahi et al., 2009).

Sánchez et al. (2009) proposes association rules for credit card fraud detection. Using data from credit card companies in Chile, they form fuzzy association rules for credit card detection and found the support and confidence of each rule. Forming of association rules were performed by k-means algorithm. They suggest that their methodology does not only improve detection results, but also facilitates understanding of alerts by fraud specialists through “linguistic labels” for alerts (Sánchez et al., 2009).

Van Vlasselaer et al. (2015) proposes enhancing transaction aggregation variables with network variables. Network variables indicates the degree of relationship between fraudulent merchants and customers and legitimate merchants and customers obtained from a network scheme. They tested the contribution of network variables and aggregated RFM (Recency, Monetary, Frequency) variables and found that inclusion of network variables significantly improves performance of fraud detection model. They also made a comparison between random forest, logistic regression and neural networks and concluded that random forest performs better than other algorithms. They use undersampling in populating random forest algorithm. Their test dataset comprises of all transactions in a week and they suggest that 90% of fraud cases were discovered with 1% false positive ratio (Van Vlasselaer et al., 2015).

Krivko (2010) proposes a one-class classification method for debit card fraud detection. His research is one of the few specifically addressing debit card fraud. His main tenet is forming “description boundaries” for each account group for a time window, which is similar to segments considering the average amounts and counts of withdrawal. In description boundaries, he discusses choosing different time windows such as 1, 3 or 7 days. He suggests 7 groups with different spending characteristics and establishes a logistic regression to measure the outlierness of the transaction within the boundaries of the group. He claims that although 7 days aggregation produce better accuracy, it could detect frauds later than shorter aggregation periods. Finally, he compares between an operational rule-based fraud detection system and hybrid model which includes his proposed one-class classification (Krivko, 2010).

Whitrow et. al (2009) compares detection two models which take either transaction level variables or aggregated variables. They conclude that aggregated variables generally perform better than transaction variables and this impact is significantly more evident in random forest algorithm. However, it should be noted that they treat aggregation in account level rather than transaction level. For this reason, relevant attributes of the transaction may not be included in the model. Like the literature mentioned above, their article suggests that performance of random forest is superior over KNN, SVM, logistic regression, CART and quadratic discriminant (Whitrow et al., 2009).

Tasoulis et. al (2008) suggest a different detection method for plastic card fraud. Their method is based on streaming data concept, which fits to movements of plastic cards through time. They use 20 variables for clustering streams of credit card movements. Two of the variables are the time of the transaction and average amount of transaction for each account. The rest of the variables represent 18 merchant sectors which can be gathered from data. Later, they use stream clustering techniques and determine the clusters in which at least one fraud is placed. It is concluded that outliers detected in stream clustering may contribute fraud detection models (Tasoulis et al., 2008).

Using Von Mises distribution, Bahnsen (2016) proposes adding “periodic mean” of transaction hours into the variables and concludes that this would improve classification results (Bahnsen et al., 2016).

Considering the literature we reviewed, a successful card fraud detection solution should take into account some topics which are mentioned below:

- Credit card fraud detection is a difficult field because of the imbalance of fraud data vis-à-vis non-fraud data. There are some methods for tackling this issue, however, they can only barely alleviate the problems associated from class imbalance.

- In relation with first feature, the false positive ratio becomes an important concern. The FP ratio not only makes it difficult to examine alerts and increases the cost of detection, and hence reduces the economic rationale of detection, but also leads discomfort in customers whose legitimate transactions are questioned.
- It is generally agreed that pure transaction data do not explain fraud adequately. For this reason, account-level or segment-level aggregation of transaction data is required. In such an aggregation process, time window, included variables and segmentation procedures, if decided, should be carefully planned and analyzed.
- Because credit card transaction data is considered secret by financial institutions, it is difficult to get adequate data for analysis. The data used by one researcher could not be compared with other researcher due to privacy concerns.
- Comparing performance of models in development stage is difficult. While some of the models employ general measures of accuracy, area under curve, true positive rate, false positive rate or precision, some others consider different misclassification cost measures such as saved limit or some cost-based interpretation of TPR and FPR. The arguments of using misclassification cost as a performance measure seems relevant, but not all algorithms are suited for such an analysis and appropriate algorithms may be difficult to apply.
- Performance of the developed algorithms are generally benchmarked against a limited test dataset and it may be difficult to interpret the results in the real-time data, which have much higher imbalance. The seemingly sound models having high test performance may fail to meet the requirements of real-time datasets or may produce extensive number of false positives which could not be detected.
- Due to the adaptive nature of customer behaviors, aka population drift, models should be frequently updated to keep their robustness.
- Some of the fraudulent transactions may remain as non-fraudulent in the database due to not having realized. Such an overlapping situation may deteriorate performance of some of the classifiers like neural networks.
- Segmentation may be considered to better understand the data and, possibly, produce different models for each segment. However, producing multiple segments and managing different models for each segment may mean more complexity for the overall solution.

3. DEBIT CARD FRAUD DETECTION THROUGH RANDOM FOREST

In our study, we will develop a classification model to determine fraudulent debit card withdrawal transactions on ATMs. In the coming section, we will firstly introduce our dataset and variables. Secondly, we will discuss separating our data between training and test datasets. Thirdly, using the variables and training/test datasets, we will establish a random forest classification model for debit card fraud detection. Finally, we will discuss the results of some experiments based on different parameters of random forest, sampling procedures and inclusion of some subset of variables. Performance metrics of the experiments will be compared. In our concluding remarks, we will discuss results of the models we established and suggest areas to work with in the future.

3.1. Rationale of Detection of Fraudulent Debit Card Transactions

As mentioned in the previous section regarding data mining, several points should be considered for establishing a fraud detection model. Although it may be tempting to establish an innovational model due to increasing popularity of data analytics solution, pros and cons of such an attempt should be evaluated.

First of all, debit card fraud is a threat for reputation of the bank in the market. The impact of that threat may aggravate due to social media outlets. Because reputation is the most important asset for financial institutions, such frauds, particularly when they are concentrated on a bank, may bring unprecedented and invisible indirect losses. This potential threat could not be easily quantified, but it should be taken into consideration.

Secondly, detection of realized fraudulent withdrawals as early as possible would prevent perpetrators to focus on the bank by being an easy prey and would minimize losses. Thus, such a measure may direct them to less-prepared institutions for their fraud attempts.

Thirdly, almost all of the loss realized due to debit card fraud is paid by the Bank. Although it is not an alarming amount, it requires some actions to keep the losses in control. The bank already applies some checks on frequency or amounts, but they are generic and thus not fed by the behavioral history of the customer. For this reason, establishing a fraud detection model for debit card frauds will help to control losses.

Finally, although establishing a real-time system may be costly, it would be beneficial to establish a preliminary classification model to see the potential offered by machine learning for fraud detection.

3.2. Dataset and Variables

Our dataset consists of debit card transactions of a bank. We have included only withdrawal transactions realized in ATM's, thus excluded POS transactions realized in merchant businesses by debit cards. Our decision is based on the consideration that so few transactions tend to be fraudulent in POS transactions of debit cards and the examination of POS transactions require different set of variables. The dataset has included transactions realized in a given year provided by the bank.

The relevant and available fields in our original dataset are;

Table 3.1. Original fields in the dataset

Account Number	The account number of debit card customer
Transaction Date	Date of the withdrawal transaction
Amount in TL	Withdrawal amount in TL
Currency	Currency of the withdrawal
Previous Balance	Balance of the account before withdrawal
Is Our Bank	Did transaction realize in our bank ATM or in different bank ATM
Age	Age of the customer
Education Level	Education level of the customer

As suggested in the relevant literature mentioned above, transaction variables or demographic variables in plastic card fraud detection does not suffice to establish a robust classification model. Previous transaction pattern of the customer, which may also be known as customer profile, should also be incorporated into the analysis for better understanding customer's current movement compared to its history (cf Bolton & Hand, 2001; Edge & Falcone Sampaio, 2009). In our analysis, we will basically use RFM (recency, frequency and monetary) variables to describe previous behavior of the customer. In this framework, Recency means time since last transaction, Frequency means number of transactions and Monetary means variables related with amounts of transactions. All of these variables can be computed for different time windows. In the credit card fraud detection literature such as in Bhattacharyya et al. (2011) or Van Vlasselaer et al. (2015), time windows varying from one hour to 3 months have been proposed, partly dependent to the availability of data. Based on these different time windows, we have determined five different time windows to describe transaction pattern of the customer: 1 hour, 1 day, 1 week, 1 month and 3 months.

RFM variable types for each customer are defined as (1) time since last transaction, (2) number of transactions, (3) total amount of transactions, (4) average amount of transactions, (5) is first withdrawal, (6) transaction amount difference from average and (7) transaction amount ratio to average.

While we speak on customer account history, it is important to understand that each account movement may have different nature. For ATM transactions, customer may have different transaction patterns in related bank ATMs, contracted ATM network, in the specific ATM in which customer have withdrawn money and in the currency withdrawn in the last transaction. Based on those differences and the insight taken from Van Vlasselaer et al. (2015), we have determined four different transaction groups for the same customer: all transactions, transactions in the same ATM, transactions in the ATM group (examined bank's ATM vs. other banks' ATM) and transactions in the same currency. Our final RFM variables are composed of cartesian product of transaction group, variable type and time window. In the next table, the details of the systematic RFM variables are given.

Table 3.2. Systematic RFM variables used in the model

Variable Type	Sub Group	Time Window
Time since last transaction (TimeSince)	All SameATM SameATMGroup SameCurrency	1 Hour/1 Day/1 Week/ 1 Month/3 Months
Number of transactions (Count)	All SameATM SameATMGroup SameCurrency	1 Hour/1 Day/1 Week/ 1 Month/3 Months
Total amount of transactions (Amount)	All SameATM SameATMGroup SameCurrency	1 Hour/1 Day/1 Week/ 1 Month/3 Months
Average amount of transactions (Average)	All SameATM SameATMGroup SameCurrency	1 Hour/1 Day/1 Week/ 1 Month/3 Months
Is first withdrawal (FirstWd)	All SameATM SameATMGroup SameCurrency	1 Hour/1 Day/1 Week/ 1 Month/3 Months
Transaction amount difference from average (DiffAverage)	All SameATM SameATMGroup SameCurrency	1 Hour/1 Day/1 Week/ 1 Month/3 Months

Transaction amount ratio to average (RatioAverage)	All SameATM SameATMGroup SameCurrency	1 Hour/1 Day/1 Week/ 1 Month/3 Months

RFM Variables are in the format of VariableType_GroupName_TimeWindow. For instance, TimeSince_All_Hour means time (in minutes) since last transaction realized in the hour prior to the current transaction (without considering ATM or currency).

In the "time since last transaction" variable type, many variables for many records could not be computed because of lack of data for particular customer. However, random forest does not handle missing values. For this reason, we made a workaround of updating those missing values as maximum amount possible in those groups. For instance, if there is no transaction in the last hour, then the value of TimeSince_All_Hour variable would be assumed as maximum available value for this variable, which is 60 minutes.

On the other hand, for many records, the value of the "Transaction amount ratio to average" variables could also not be computed, because the average would be 0. In such situations, this ratio is assumed to be the same as the current transaction amount. For instance, if TimeSince_All_Hour is missing, RatiotoAverage_All_Hour variable would be equal to transaction amount.

Considering all these variables, there will be 7 variable types and 4 groups for each variable type against 5 windows of time. The total number of systematic RFM variables would be 7x4x5=140.

Beside those systematic RFM variables, following ad-hoc RFM variables are also produced with consideration of the relationship of the current transaction to the whole set of previous transactions of the customer.

Table 3.3. Ad-hoc RFM variables

Largest Transaction	Largest transaction amount of the account before this one
Time Since Last Transaction	Time (minutes) passed since last transaction (If N/A account opening date)
Withdrawal Ratio	Ratio of the current withdrawal to minimum of previous balance or largest transaction

In total, 149 variables are included in our analysis. 140 of them are systematic RFM variables introduced before, 3 are ad-hoc RFM variables, 6 are transaction variables mentioned above (excluding account number and transaction date).

Time-based separation of training and test datasets is used in such articles as Baesens et al. (2015) and Bhattacharrya et al. (2011). The idea is that out-of-time testing would be a robust way of determining performance. In our analysis, we have divided the dataset to training and test dataset based on two different time periods, thus it is ensured that no overlapping occurs between them. The datasets are quite unbalanced having fraudulent records less than % 0.1 of the total. Details regarding training and test datasets are given below:

Table 3.4. Information about training and test datasets

	Training	Test
Frauds	331	257
Non-Frauds	435.234	339.605
Total	435.565	339.862
Ratio	% 0.076	%0.076

3.3. Choosing Classification Algorithm and Performance Measure

In our thesis, we aim at establishing a classification model for fraudulent ATM cash withdrawals. In this section, we will discuss considerations in selection of appropriate classification algorithm and performance measure.

3.3.1. Choosing the Appropriate Classifier for Card Fraud Detection Problem

As we have partially explained in literature review section, there are many classification algorithms used for card fraud detection problem. A non-exhaustive list would include a wide-variety of methods such as decision tree methods (C4.5, CART), one-class classification methods, ensemble methods, neural networks and Fischer discriminant. Having such a variety of algorithms, choosing an appropriate one for our classification task requires considering several criteria.

First of all, as a rule of thumb, there is no single algorithm that fits all problems. For this reason, an algorithm working fairly in one problem would not be appropriate in other problems. However, there are numerous studies performed in the card fraud detection field and there are evidence that random forest algorithm performs better than SVM or logistic regression, which may be treated as “baseline” classifiers (Bhattacharyya et al., 2011; Dal Pozzolo et al., 2015; Van Vlasselaer et al., 2015; Whitrow et al., 2009). Random forest is also known for its resistance against overfitting.

The second consideration is the complexity of the algorithm. The literature reviews states that the usage of more complex and advanced classifiers are limited in the field

(Bhattacharyya et al., 2011), which may be the logical conclusion of Occam's Razor principle. For data mining problems, this principle suggests that "use the least complicated algorithm that can address your needs and only go for something more complicated if strictly necessary" (Amatriain, 2015). For this reason, it is reasonable to start from relatively simple algorithms. In that sense, random forest algorithm is the ensemble created from CART decision trees and, for this reason, quite straightforward for understanding. Its parameters are not complex and there are some preset values which generally produce acceptable results.

Another important factor for the choice of random forest algorithm is its relative tolerance to variable problems frequently observed in simple algorithms such as logistic regression, such as lack of linearity of variable values or interactions. This is important in our case, because, due to our imputation procedure, requirement of linearity would render solution of our problem impossible.

In terms of interpretability, random forest provides ranking of variable importance. In fact, this is one of the most favorite features of random forest. Although it does not reveal the interaction direction, understanding variable importance ranking facilitates comprehension of classification algorithm.

Random forest algorithm used with behavioral variables (like RFM variables) has also the advantage of easily adapting population drift. Behavioral variables, when crafted carefully, adequately reflect changing behaviors of customers and easy training of random forest enables incorporating the changing nature of fraud detection into model.

Random forest's scoring mechanism has the benefit of leveraging false positive rates to conform a specified "detection appetite". Through this benefit, it is possible to differentiate actions based on the classification score and this is another important advantage of random forest.

For our problem, handling unbalanced data is also an important choice criterion. Random forest has mechanism to work quite successfully with unbalanced data through its stratified sampling option.

Speed is another key concern in algorithm selection. Training and prediction speeds should be adequate for the problem at hand. In a runtime implementation of prediction, random forest is found slower than neural networks, but this can be remediated by fine-tuning number of variables and number of trees (NoName, 2016). In terms of speed or performance, we expect our algorithm to be a baseline algorithm, rather than a fully-fledged integrated solution. For this reason, speed is not our instant priority. However, the algorithm is suitable for fine tuning for adequate speed when required.

Due to the factors mentioned above, we have chosen random forest algorithm to predict debit card fraud. The proven performance of the algorithm in the literature review and the nature of our variables together with the simplicity of implementation became the most prominent drivers of our choice.

3.3.2. Determining Performance Measures

To compare model performances, many metrics were proposed in the literature. Metrics having variable misclassification costs would be more appropriate for our problem, but we could not use them because random forest algorithm could not directly employ them. As mentioned in the literature section, accuracy measure could not be used in such a highly imbalanced dataset. In our training dataset, if a classifier labels all transactions as non-fraudulent, it would reach an accuracy rate of 99.92%, which suggests a highly accurate model, but does not give us any information for true positive records. If our classifier would use variable misclassification in fitting the model, then discovered fraud amount or total saved account balance (cf. Mahmoudi & Duman, 2015) could be used as a performance metric, but that is not available in our settings. Area under curve (AUC) is generally accepted as a performance metric in such imbalanced-dataset problems. Secondly, Vlasselaer et al. (2015) suggests that true positive rate (TPR) achieved by 1% examination rate in test dataset may be a performance metric. We propose that instead of TPR in examination rate of 1% of dataset, TPR reached in examination of 10 times of number of frauds may be a more comparable measure. For instance, if the number of fraud cases in test dataset is 250, models can be compared based on true positive rate reached in the top 2500 records ordered by classification score. Thus, this measure can be used to partially contain false positive rate while indicating true positive rate. Some measures and statistics that are used in comparing model performances are listed in the following table.

Table 3.5. Performance statistics of random forest models

Statistics	Explanation
ACC	Accuracy rate (TP+TN/TP+TN+FP+FN)
AUC	Area under receiver operating curve
TPR	True positive rate (TP/TP+FN)

TPR10F	True positive rate found in 10 times number of fraud records

Random forest produces different outcomes with same parameters if the random seed value is set as different. Due to this randomness, it may be considered that running each test more than one and averaging performance metrics would produce more comparable results. This approach is used in Mahmoudi & Duman (2015) to compare different models with averages of model results for three runs. Along with this insight, we run each test five times, with different seed values, to compare their results. Standard deviation of true positives detected in the 10*Fraud observations were also given, but it may be erroneous to reach conclusion based on the level of standard deviation, because each test was conducted five times, which may not be enough for drawing conclusions.

3.4. Experiments on Random Forest Classification Model

For implementation of random forest classification in our thesis, we have used RandomForest formula of RandomForest package in R software (Liaw, 2015). The syntax and parameters of RandomForest formula is given in the Appendix-1.

In order to determine the robustness of the model and to compare the impact of parameter changes, we established an experimental setup using R software and SQL Server 2016. In this setup, a table was filled with the random forest parameters which will be tested. Through an R script, a loop was created which takes these parameters one by one and fits random forest model with those parameters, five times for each row. If a percentage of the training dataset is taken into account, training dataset is redrawn with replacement for each test. The performance metrics mentioned above was calculated for each test and those metrics along with variable importance results are stored in tables. In the following table, parameters tested in our experimental setup are listed:

Table 3.6. Parameters used in experimental setup

Parameter Name	Explanation
Training percentage	Random percentage of training dataset records included in the analysis.
Number of trees	Number of trees taken in random forest algorithm.
Number of variables	Number of variables over all variables taken in each tree.
Subsampling-NF Class	For subsampling (stratified sampling in each tree), sampling percentage of non-fraud cases over all fraud cases.
Subsampling-F Class	Sampling percentage of fraud cases over all fraud cases for subsampling.

3.4.1. Impact of Training Dataset Size and Subsampling

In our first experiment, we have explored the impact of changes in training dataset size in the results. We have not used subsampling capability of random forest for these tests. We have used randomly chosen 1%, 10% and 20% of training datasets to see the impact of changes of training dataset size. Actually, we also used 50% of the training dataset, but the hardware configuration of our testing machine produced an error for that size. Number of trees and variables are kept in their default values, 500 and 12, respectively. The results are given in the following table.

Table 3.7. Impact of changing training percentage on RF model performance

Training Percentage	TP Standard Deviation in 10F	Cutoff Point 10F	AUC	TPR	TPR10F	ACC
1	4.16	56%	97.5%	83.1%	79.9%	99.1%
10	2.92	20%	97.2%	51.9%	80.5%	99.8%
20	1.87	13%	96.9%	39.7%	80.2%	99.9%

Results show that, without subsampling, increasing training percentage substantially reduce true positive rate. It is understood that sensitivity reduces against specificity while increasing training percentage, and thus increasing imbalance ratio. However, interestingly, TPR10F measure seems robust for such changes. While increasing training percentage, standard deviation in TP found in 10F reduces but the TPR10F measure, on average, seems stable. In these tests, accuracy remains so high because of lack of subsampling, and gets higher while increasing training percentage included in the analysis.

In the second test, we have applied subsampling and choose a ratio of 250:100 between non-fraud and fraud observations in each tree. That means each tree would get constructed considering all fraudulent observations (100%) and non-fraudulent observations which are 2.5 times of fraudulent observations (250%). Other parameters were kept same (number of trees are 500 and number of variables in each tree is 12). In this test, percentages considered in training dataset, which have 435.000 records, were also set as 1%, 10% and 20%.

Table 3.8. Impact of changing training percentage with subsampling

Training Percentage	TP Standard Deviation in 10F	Cutoff Point 10F	AUC	TPR	TPR10F	ACC
1	4.76	67.4%	97.6%	87.4%	78.9%	98.1%
10	1.30	64.5%	97.6%	86.5%	80.6%	98.3%

20	0.44	65.2%	97.7%	86.4%	80.2%	98.2%

It is evident that subsampling has led more stable results in terms of true positive rate and area under curve measures. The measures do not deviate so much and sensitivity (true positive rate) reduced only slightly while increasing training dataset size as opposed to the picture in the previous test settings. TPR10F measure seems to be stable and using 10% of the training dataset produced slightly better results. Standard deviation of true positives in 10*Fraud observations reduced significantly in 20% of dataset. Using a larger training data may produce more stable outcome regarding TPR10F, while it reduced the rate slightly. Accuracy reduced compared to tests without subsampling due to the fact that subsampling lead more false positives. Due to the fact that slightly best TPR10F measure results were obtained using 10% of training dataset in the previous two tests, we will use 10% of training dataset in our following tests. In terms of subsampling, we will use 250:100 subsampling ratio as non-fraud/fraud ratio (NF/F).

3.4.2. Impact of Number of Trees and Number of Variables

Number of trees and number of variables are other parameters that may have impact on the model results. According to Liaw&Wiener, "The number of trees necessary for good performance grows with the number of predictors". Hence, a dataset having 149 variables should be analyzed with a relatively high number of trees. For selecting number of predictor variables, following the idea of Prof. Breiman, who discovers random forest algorithm, Liaw & Wiener (2002) suggests that half of the default or twice of the default number of variables may be tried at first to experiment whether number of variables change the performance of the model. However, it is also stated that number of variables do not generally improve model results and even setting the variable number as 1 may produce acceptable results (Liaw & Wiener, 2002).

For our dataset, we have tried different number of trees and number of variables and compared the results. Training percentage is set as 10% in these experiments. Stratified sampling was also implemented with 250:100 ratio for non-fraud/fraud cases.

Table 3.9. Impact of changes in number of trees and variables on performance

# of Trees	# of Variables	TP Standard Deviation in 10F	Cutoff Point 10F	AUC	TPR	TPR10F	ACC
500	12	1.30	64.5%	97.6%	86.5%	80.6%	98.3%
50	12	5.92	66.8%	97.5%	87.0%	78.2%	98.1%
5000	12	1.00	65.0%	97.7%	86.2%	80.2%	98.3%

500	6	1.64	65.8%	97.6%	86.6%	79.8%	98.2%
500	24	1.34	66.0%	97.6%	87.0%	79.6%	98.2%

The results suggest that increasing or reducing the number of variables in each tree seems not contributing the performance of the model. For reducing number of trees, it is clear that it reduces the performance and increase observed difference between results for each test. Overall, it may be expected that increasing the number of trees may bring more stability to test results, but it did not contribute to the performance in terms of TPR10F measure. Accuracy and AUC measures seems not affected from these changes as well.

3.4.3. Impact of Subsampling Parameters

Unbalanced datasets are a challenge for establishing robust models in most of the algorithms, including random forest. There are some measures which can be used to reduce the impact of imbalance issue in random forest, but the most accepted practice is subsampling/stratified sampling (NoName, 2015).

As mentioned above, each tree in the random forest normally takes around 2/3 of determined training data to fit the model. In unbalanced datasets, that means lack of adequate learning from rare class cases in each tree formation. Subsampling (generally) takes all rare class members and a proportionate number of frequent class observations in the training of each tree. Thus, it is mostly used as a form of undersampling, as discussed in the data mining section. In Vlasselaer et al. (2015), it is suggested that two times of fraud cases should be taken from non-fraud cases. In line with this insight, we have used the ratio of 250:100 for non-fraud/fraud cases in subsampling and used this in our tests as benchmark.

In this section, we will compare various subsampling ratios and examine the performance impact of them. For that test, we take base scenario and other scenarios having various subsampling ratios. In the following table, results of our tests are presented.

Table 3.10. Performance results of various subsampling ratios

NF/F Sampling	TP Standard Deviation in 10F	Cutoff Point 10F	AUC	TPR	TPR10F	ACC
250:100	1.30	64.5%	97.6%	86.5%	80.6%	98.3%
100:40	2.51	73.1%	97.6%	87.6%	74.2%	97.5%
100:100	1.14	73.9%	97.8%	91.0%	77.6%	96.6%
300:100	2.77	63.2%	97.6%	85.2%	80.5%	98.4%
600:100	2.62	54.8%	97.6%	83.8%	81.4%	99.1%
1000:100	1.82	48.5%	97.6%	81.2%	81.9%	99.3%
1500:100	1.64	43.3%	97.4%	78.1%	82.2%	99.5%
2000:100	1.64	39.1%	97.5%	74.2%	82.0%	99.6%
5000:100	2.64	28.0%	97.5%	62.3%	81.7%	99.8%

As opposed to the argument stated by Vlasselaer et al. (2015), there seems a clear improvement in TPR10F performance measure with the increase of not-fraud/fraud (NF/F) ratio in subsampling until the ratio of 1500:100. From this level, the ratio starts deteriorating slightly. As we have seen in our first test, where we try to determine optimum level of training percentage in each test, accuracy improves while true positive rate deteriorates due to increasing NF/F ratio in each tree. The worst performance above was observed when subsampling limits fraud cases in each tree. This finding further advocates subsampling through taking all fraud cases. However, the optimum ratio of subsampling remains a research issue although higher ratios than literature were found beneficial for performance.

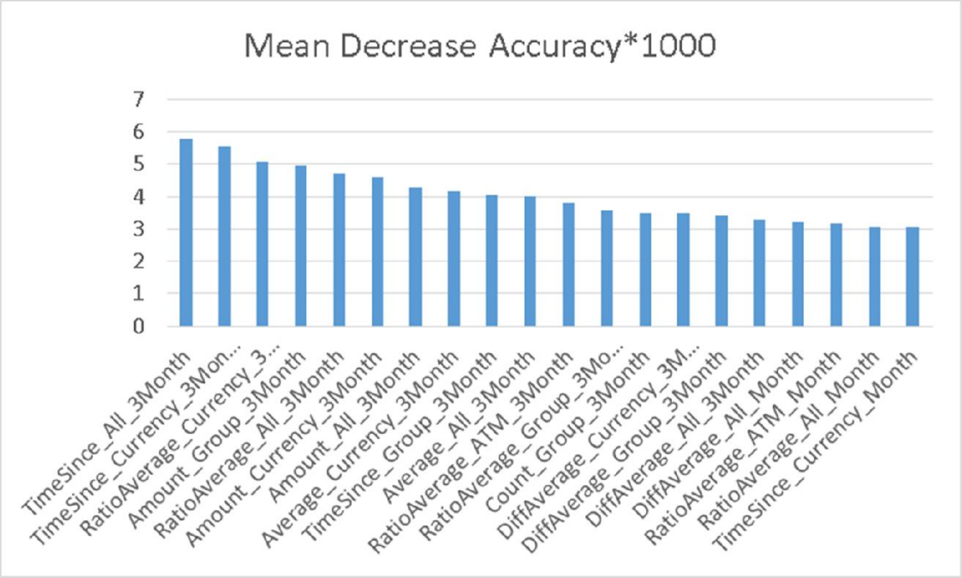
3.5. Importance of Variables

One of the most important, and frequently used, features of random forest is giving variable importance information. This information is given in two measures: Mean decrease in accuracy and mean decrease in Gini (node impurity) when a variable is removed from predictors. According to Liaw & Wiener (2002), "The random forest algorithm estimates the importance of a variable by looking at how much prediction error increases when (OOB) data for that variable is permuted while all others are left unchanged." While importance measures may differ from one run to another, the rank of the variables is expected to be same. On the other hand, it should be noted that importance of a variable, as calculated by random forest algorithm, may be the result of its interaction with other variables (Liaw & Wiener, 2002).

It is stated that mean decrease in Gini is biased in favor of variables having many categories and continuous variables. Mean decrease in accuracy, on the other hand, is unbiased when subsampling is used (Strobl & Zeileis, 2008). In this study, we will use mean decrease in accuracy as the main indicator of variable importance.

For our base scenario, (500 trees, 10% training data, 12 number of variables, 250:100 subsampling), average of mean decrease in accuracy for each predictor variables are given in Appendix 2. Top 20 important variables are given in the following figure.

Figure 3.1. 20 most important variables found by the algorithm

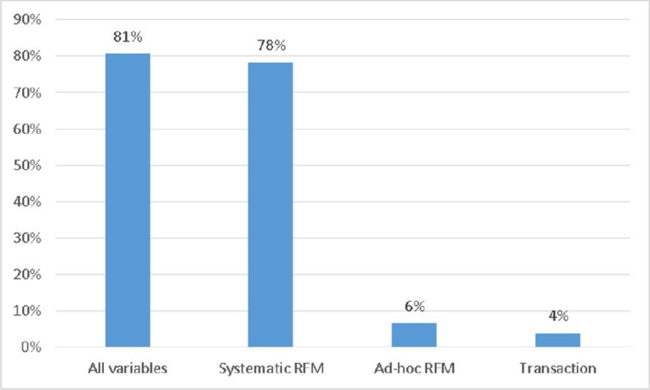


In the top 20 variables, it is evident that month and 3 months variables have more prominence. Furthermore, in this list, there are no variables except systematic RFM variables. This figure emphasizes the importance of systematic RFM variables in our analysis.

On the other hand, some of the variables have negative signs with regard to “mean decrease in accuracy” measure. For this reason, we have truncated variables having negative sign in any of the tests in our base scenario and fit an RF model with remaining variables. Interestingly, no improvement was observed in performance metrics vis-à-vis base scenario.

To understand the impact of our variables, we ran our analysis considering different variable groups and compared the results. Firstly, we compared the main variable groups in our variable list: systematic RFM variables, ad-hoc RFM variables and transaction variables. As performance metric, we used TPR10F measure. The results are presented in the following chart.

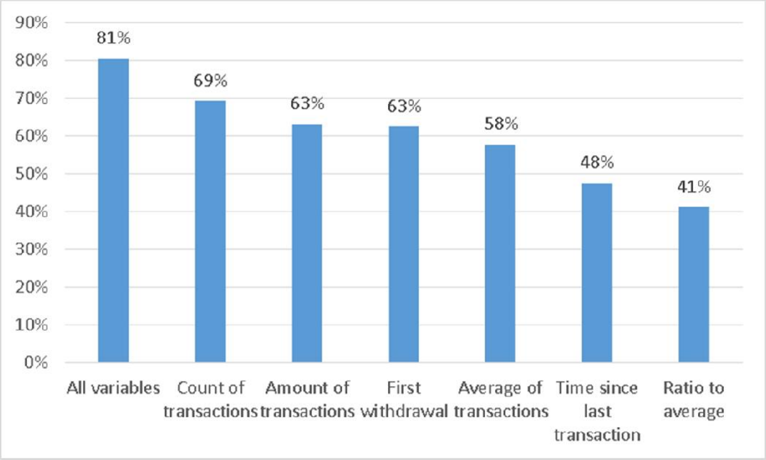
Figure 3.2. Performance of RF models (TPR10F) with main variable groups



As can be seen, the most significant variable group is systematic RFM variables. They can detect 78% of fraud cases without ad-hoc RFM variables and transaction variables. Transaction variables, which are the only variables at our hand before aggregation of customer-based debit card transactions, could only detect 4% of fraud cases in 10*Fraud observations. This chart emphasizes the importance of aggregation of transactions for debit card fraud detection.

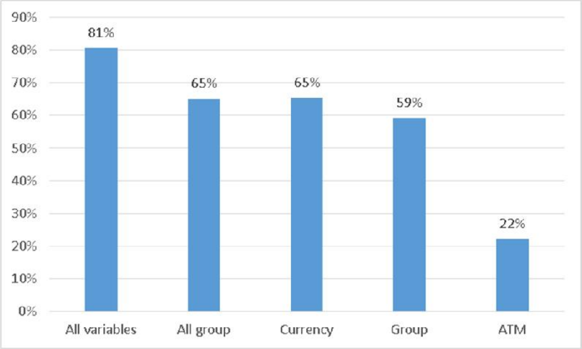
Later, we have made a decomposition of systematic RFM variables in terms of (1) variable type, (2) group and (3) time window. Like the analysis above, random forests were fit with determined groups of systematic RFM variables and performance results (TPR10F) are presented below.

Figure 3.3. Performance of RF models (TPR10F) with systematic RFM variable types



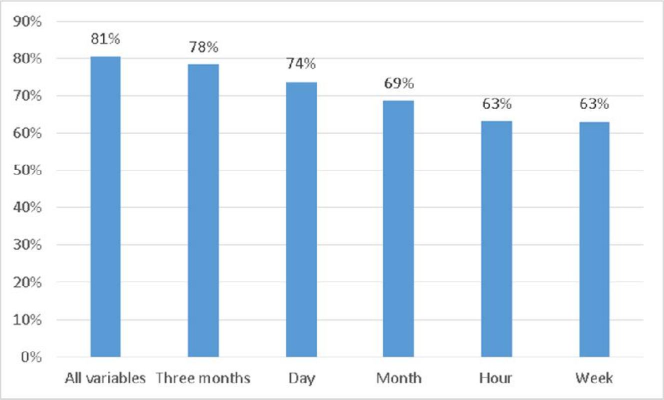
As can be seen from above chart, systematic RFM variable groups have relatively close performance results. If we made a model containing only count of transactions variables (20 variables), the model would discover 69% of fraud cases with 10*F observations. The proximity of values for groups suggests that many of our variables may indicate similar fraud cases. In the following chart, same analysis is made for sub-groups in our systematic RFM variables.

Figure 3.4. Performance of RF models (TPR10F) with RFM variable sub-groups



In terms of sub-groups, it is seen that models fitted by considering all transactions and considering only same currency transactions have same performance results. This may partly because of the fact that most of the currency values in the dataset are denominated in Turkish Lira and thus having same currency may almost mean having all transactions. Group variables in their own could discover 59% of fraud cases, while transactions conducted in the same ATM group could explain 22%. This finding may indicate that making transaction in the same ATM could only partially contributes to the overall analysis of fraud detection. Finally, we compared performance of models which include different time windows.

Figure 3.5. Performance of RF models in different time windows



As we have already seen in the first 20 important variables, 3 months variables are the most important variables. An RF model fitted with three months variables (28 variables) could discover 78% of fraud cases within 10*Fraud observations. This is surprising, because, an RF model fitted with a model including only three months variables perform at par with all

RFM variables and quite close to the model with all 149 variables. This may be the result of correlated variables and should be further investigated.

4. Conclusion

We have applied several tests on random forest model established for debit card fraud detection. Our tests hint several points that should be taken into account in the solution of such problems. In this section, we will summarize most significant findings we reached in our analysis and discuss issues that need to be considered. We will also propose emerging research questions in similar topics.

4.1. Training Dataset Size and Formation

Although it may be argued that the higher training dataset size is bigger, our results show that the gain obtained from larger training dataset is quite limited. In our experiments, 10% sample of training dataset produced slightly better predictive performance against 20% sample. Higher training dataset sizes created problems emanating from inadequate computational resources. Due to the comparable performance and advantages of calculation time, we choose 10% sample (43.556 observations) for training dataset size. However, this dataset was replaced with a new training dataset for each 5 trials performed in each test.

In our analysis, we have separated training and test datasets based on a certain time point. However, it may also be thought that dataset separation could be made based on customers. For instance, training datasets may include transactions of 1000 customers, of which hundreds have fraudulent transaction while test dataset may include 1500 customers of which 120 have fraudulent transactions. In our study, no experiment was made based on such a separation, but it may be a future research topic.

4.2. Stratified Sampling

Treatment of unbalanced datasets is important for establishing robust models appropriate to such data. One way of unbalanced dataset treatment in random forest is stratified sampling in the training dataset in each tree formation. In our analysis, we have concluded that subsampling up to 15/1 in each tree would improve model performance. After that level, subsampling up to 50/1 in each tree seems not further contributing model performance. The other methods of imbalance treatment, like different class weights, were not considered in this analysis and may be a research topic.

4.3. RFM Variables and Variable Importance

In terms of variables, we concluded that systematically produced RFM variables may be used to incorporate account history of the customers for such analyses. The usage of systematically produced RFM variables (140 variables) in our analysis could detect 78% of fraud cases in 10*Fraud observations.

One usage of random forest is determining importance of variables. However, random forest model sometimes is perceived as a black-box model because it does not explain the interaction of variables for model results. In our analysis, although variable importance in detail is given in Appendix-2, we could not determine whether these variables increase or reduce fraud tendency. Including logistic regression to the analysis may be helpful to explain the direction of impact for predictive variables. However, due to our imputation, logistic regression is expected to confront difficulty of assigning coefficients to variables.

In our analysis, we used time windows for determining RFM variables. Instead of time windows, it may worth performing research to use last n number of transactions of the customer to define his/her account history. The main setback of time windows is the fact that many customers do not have transaction in the determined time window, especially for shorter time windows, and this situation requires large-scale missing value treatment, which may have negative consequences on the model robustness and performance. Some kind of combinations may also be considered for defined time windows and last n number of transactions.

For missing values, which are frequently observed in our systematically produced RFM variables, we have made some assumptions and imputed them. However, due to their high frequency, that process created so many highly correlated variables. Although random forest model can handle correlated variables, model robustness may be impaired because of so many correlated variables. For that reason, impact of large-scale imputing of missing values should be explored in future studies.

In our analysis, RFM variables for 3 months period (28 variables) could indicate same level of performance (78% TPR10F) as all of the systematical RFM variables. It may be argued that this situation may be the result of random forest's tendency to prioritize variables having more categories (Bhalla, 2014). (In our dataset, variable group having least amount of missing values in initial form is 3 months period variables. Hence, the number of categories is expected to be more than any other groups for three months.) However, if we look at the second most important time window, we see that it is daily time window. Thus, this issue seems to be the result of correlation produced by either large-scale imputing of missing values or variables measuring same thing again and again.

4.4. Other Random Forest Implementations and Parameters

Although we have changed several important and frequently used parameters of random forest formula, we did not make experiments on all parameters. As can be seen in the Appendix-1, RandomForest formula have different parameters that enable (1) replacing or not replacing withdrawn sample each time, (2) including prior distribution of classes/class weights, (3) determining minimum size of terminal nodes, default of which is 1 for classification and 5 for regression, (4) determining maximum number of terminal nodes. These parameters may also have impact over the result of the analysis and this is also a research field in similar topics. Furthermore, there are other packages in R software that can handle random forest analysis with different features, such as party/cforest and ranger. Those packages with their different features could also be explored in future researches.

4.5. Robustness and Performance Metrics of Random Forest Models

Because of the randomness inherent in random forest, the performance results may differ somewhat for each run of the model. Because of this, it is important to determine the robustness of the fit before drawing further conclusions. In our analysis, we have included the standard deviation of true positives reached in 10*Fraud observations as a measure of difference for each run of the test with the same parameters. In general, this measure can indicate the robustness of the model, especially when statistically significant numbers of tests are conducted. It may worth further research to determine a measure of robustness for a random forest model.

In this thesis, we tried a different performance metric for problems involving unbalanced datasets. In our opinion, true positive rates found in 10*Fraud observations may be a comparable measure when different parameters are tried, a practice which changes the observations in the cutoff threshold. For instance, when subsampling is not implemented, true positive rates are found to be so low in cutoff value of 0.5. Any metric including true positive rates such as balanced accuracy would produce erroneous results in such cases. However, if we consider true positive rates found in 10*Fraud observations (TPR10F) as performance metric, we conclude that even in rare class problems, random forests without subsampling indicate comparable, albeit somewhat lower, performance.

TPR10F measure can also be used as a measure of “detection appetite”. It shows that the practitioner accepts examining 10 times of fraud cases in order to detect certain percentage of fraud cases. This ratio can be changed for different detection appetites. Furthermore, this ratio can also be used to differentiate actions that will be implemented considering the riskiness of transaction. For instance, in a real-time debit card fraud detection setting, a classification model giving the highest TPR3F measure can be implemented as stopping the transaction and calling customer. If this test fails, a model having highest TPR10F measure can be used to send SMS to the customer for fraud suspicion and further scrutiny of the transaction. The advantage of this approach over using scores as cutoff for different monitoring practice would be acknowledging in advance possible false positive rates.

In our analysis, we concluded that TPR10F explains performance better than true positive rate, accuracy and area under curve measures. However, as evident in some of the literature including Mahmoudi & Duman (2015), variable cost based performance measures may be a better choice in problems involving monetary losses. We could not use variable cost based misclassification due to the limitations of random forests. However, random forests for regression (with fraud amount being target variable) could be considered for this purpose in future researches. This may also be evaluated on the basis of “expected loss” framework regarding case examination as suggested by Baesens et al. (2015).

4.6. Practical Implementation of Debit Card Fraud Detection

Practical implementation of a random forest model in a real-time debit card fraud detection system has some challenges. Most evident of them is real-time calculation of customer-based aggregate variables. To solve this issue, it may be proposed to run a daily task to calculate required aggregates. It may also be considered to determine limits for daily/hourly transactions in the following day using random forest algorithm.

In terms of practical implementation of fraud detection system, we could not consider existing fraud alerts in our analysis for several reasons. If existing fraud alerts are known, they may be included in the analysis as new variables, or they may be regarded as a first level defense, as seen in the literature (Krivko, 2010). This action may enhance the performance of overall fraud detection program.

Another action that could be taken to improve model performance in real-time fraud detection setting may be examining outlier records and fraud cases not detected by the model. If the reason of not being detected could be discovered, new variables can be incorporated into the model which takes into account those observations. Removing outlier records may also be beneficial.

The final consideration for real-time debit card fraud detection system is cost-benefit analysis. As mentioned in the data mining section, costs of establishing a fraud detection system should be carefully considered against benefits gained by the system. Furthermore, encouraging more secure forms of cash withdrawal, like EVM chip or, better, contactless payment should also be assessed. On the other hand, indirect impacts of debit card fraud in terms of loss of customer confidence or litigation should also be taken into account.

REFERENCES

ACFE. (2016). What is Fraud? Retrieved August 17, 2016, from http://www.acfe.com/fraud-101.aspx

Ahmed, M., Naser, A., & Islam, R. (2016). A survey of anomaly detection techniques in financial domain. *Future Generation Computer Systems*, *55*, 278–288.

Akoglu, L., Tong, H., & Koutra, D. (2015). *Graph based anomaly detection and description: A survey*. *Data Mining and Knowledge Discovery* (Vol. 29).

Albrecht, C. C., & Albrecht, C. O. (2013). Data-Driven Fraud Detection Using Detectlets. *Journal of Forensic & Investigative Accounting*, *Vol.1*(1), pp.1-24.

Amatriain, X. (2015). What are the advantages of different classification algorithms? Retrieved August 10, 2017, from https://www.quora.com/What-are-the-advantages-of-different-classification-algorithms

Argyriou, E. N., Sotiraki, A. A., & Symvonis, A. (2013). Occupational Fraud Detection Through Visualization. *ISI*, 4–7.

Baesens, B., Vlasselaer, V. Van, & Verbeke, W. (2015). *Fraud Analytics Using Descriptive, Predictive, and Social Network Techniques: A Guide to Data Science for Fraud Detection*. Wiley.

Bahnsen, A. C., Aouada, D., Stojanovic, A., & Ottersten, B. E. (2016). Feature engineering strategies for credit card fraud detection. *Expert Syst. Appl.*, *51*, 134–142.

Bhalla, D. (2014). Random Forest in R : Step by Step Tutorial. Retrieved August 20, 2017, from http://www.listendata.com/2014/11/random-forest-with-r.html

Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011). Data mining for credit card fraud: A comparative study. *Decision Support Systems*, *50*(3), 602–613.

Bolton, R. J., & Hand, D. J. (2001). Unsupervised Profiling Methods for Fraud Detection. *Proc. Credit Scoring and Credit Control VII*, 5–7.

Brockett, P. L., Derrig, R. A., Golden, L. L., Levine, A., & Alpert, M. (2002). Fraud classification using principal component analysis of RIDITs. *Journal of Risk and Insurance*, *69*(3), 341–371.

Cahill, M. H., Lambert, D., Pinheiro, J. C., & Sun, D. X. (2002). DETECTING FRAUD IN THE REAL WORLD. In *Handbook of Massive Data Sets*.

Chan, P. K., Fan, W., Prodromidis, A. L., & Stolfo, S. J. (1999). Distributed data mining in credit card fraud detection. *IEEE Intelligent Systems and Their Applications*, *14*(6), 67–74.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, *41*(3), 15.

Cortes, C., & Pregibon, D. (2001). Signature-Based Methods for Data Streams. *Data Mining and Knowledge Discovery*, *5*(3), 167–182.

Çetin, U. (2013). *Satın Alma ve Ödeme Döngüsü Hileleri ve Ortaya Çıkartılması*. Marmara Üniversitesi.

Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., & Bontempi, G. (2015). Credit card fraud detection and concept-drift adaptation with delayed supervised information. In *Neural*

Networks (IJCNN), 2015 International Joint Conference on (pp. 1–8). IEEE.

Dorronsoro, J. R., Ginel, F., Sgnchez, C., & Cruz, C. S. (1997). Neural fraud detection in credit card operations. *IEEE Transactions on Neural Networks*, *8*(4), 827–834.

Duman, E., & Ozcelik, M. H. (2011). Detecting credit card fraud by genetic algorithm and scatter search. *Expert Systems with Applications*, *38*(10), 13057–13063.

Edge, M. E., & Falcone Sampaio, P. R. (2009). A survey of signature based methods for financial fraud detection. *Computers & Security*, *28*(6), 381–394.

Eifrem, E. (2016). Graph databases: the key to foolproof fraud detection? *Computer Fraud & Security*, *2016*(3), 5–8.

Ernst & Young. (2016). Corporate Misconduct — Individual Consequences. Retrieved February 10, 2017, from https://webforms.ey.com/Publication/vwLUAssets/EY-corporate-misconduct-individual-consequences/$FILE/EY-corporate-misconduct-individual-consequences.pdf

Fawcett, T., & Provost, F. (1999). Activity monitoring: Noticing interesting changes in behavior. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 53–62). ACM.

Ferreira, P., Alves, R., Belo, O., & Cortesão, L. (2006). Establishing Fraud Detection Patterns Based on Signatures. In P. Perner (Ed.), *Industrial Conference on Data Mining* (pp. 526–538). Berlin, Heidelberg: Springer Berlin Heidelberg.

Gündüz, A. (2014). *İşletme çalişanları tarafindan yapılan hileler ve çalışan hilelerinin tespitiyle ilgili bir uygulama*. Okan Üniversitesi.

Halvaiee, N. S., & Akbari, M. K. (2014). A novel model for credit card fraud detection using Artificial Immune Systems. *Applied Soft Computing*, *24*, 40–49.

Han, J., Kamber, M., & Pei, J. (2011). *Data mining: concepts and techniques*. Elsevier.

Inc, T. (2017). EMV Chip Card and Magnetic Stripe: Card Security. Retrieved from https://pos.toasttab.com/blog/emv-chip-card-magnetic-stripe-card-security

Institute of Internal Auditors. (2016). Glossary. Retrieved July 30, 2016, from https://na.theiia.org/certification/Public Documents/Glossary.pdf

Jans, M., Lybaert, N., & Vanhoof, K. (2006). Data Mining for Fraud Detection : Toward an Improvement on Internal Control Systems ? *International Research Symposium on Accounting Information Systems*, 1–17.

Jha, S., Guillen, M., & Westland, J. C. (2012). Employing transaction aggregation strategy to detect credit card fraud. *Expert Systems with Applications*, *39*(16), 12650–12657.

Kenneally, S., President, V., Policy, C., Yao, J., & President, S. V. (2016). ABA Deposit Account Fraud Survey.

Kossovsky, A. E. (2014). *Benford's Law: Theory, The General Law Of Relative Quantities, And Forensic Fraud Detection Applications*. New Jersey: World Scientific.

Krivko, M. (2010). A hybrid model for plastic card fraud detection systems. *Expert Systems with Applications*, *37*(8), 6070–6076.

Lazarević, A., Srivastava, J., & Kumar, V. (2004). Data Mining for Analysis of Rare Events : A Case Study in Security , Financial and Medical Applications. In *PAKDD-2004* (pp. 1–54).

Lee, W., & Stolfo, S. J. (2000). A framework for constructing features and models for intrusion detection systems. *ACM Transactions on Information and System Security*, *3*(4), 227–261.

Liaw, A. (2015). Breiman and Cutler's Random Forests for Classification and Regression. Retrieved August 30, 2017, from https://cran.r-project.org/web/packages/randomForest/randomForest.pdf

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R News*, *2*(3), 18–22.

Little, B., Rejesus, R., Schucking, M., & Harris, R. (2008). Benford's Law, data mining, and financial fraud: a case study in New York State Medicaid data. *Data Mining IX: Data Mining, Protection, Detection and Other Security Technologies*, *40*, 195–204.

Longadge, R., Dongre, S. S., & Malik, L. (2013). Class imbalance problem in data mining: review. *International Journal of Computer Science and Network*, *2*(1), 83–87.

Mahmoudi, N., & Duman, E. (2015). Detecting credit card fraud by modified Fisher discriminant analysis. *Expert Systems with Applications*, *42*(5), 2510–2516.

Martens, D., de Fortuny, E. J., & Stankova, M. (2013). Data mining for fraud detection using invoicing data. A case study in fiscal residence fraud. *University of Antwerp, Working Papers*, *32*(0).

Mosley, L. (2013). *A balanced approach to the multi-class imbalance problem*. Iowa State University.

Motoda, H., & Liu, H. (2002). Feature selection, extraction and construction. *Communication of IICM (Institute of Information and Computing Machinery, Taiwan) Vol*, *5*, 67–72.

NCR. (2016). Six Types of ATM Attacks and Frauds. Retrieved August 17, 2017, from https://www.ncr.com/company/blogs/financial/six-types-of-atm-attacks-and-fraud

NoName. (2015). Random forest in R using unbalanced data. Retrieved August 10, 2017, from https://stats.stackexchange.com/questions/168415/random-forest-in-r-using-unbalanced-data

NoName. (2016). Speed of prediction: neural network vs. random forest? Retrieved August 25, 2017, from https://stats.stackexchange.com/questions/215970/speed-of-prediction-neural-network-vs-random-forest

Panigrahi, S., Kundu, A., Sural, S., & Majumdar, A. K. (2009). Credit card fraud detection: A fusion approach using Dempster–Shafer theory and Bayesian learning. *Information Fusion*, *10*(4), 354–363.

PwC. (2016). Global Economic Crime Survey. Retrieved January 15, 2017, from http://www.pwc.com/gx/en/economic-crime-survey/pdf/GlobalEconomicCrimeSurvey2016.pdf

Quah, J. T. S., & Sriganesh, M. (2008). Real-time credit card fraud detection using computational intelligence. *Expert Systems with Applications*, *35*(4), 1721–1732.

Sabau, A. S. (2012). Survey of clustering based financial fraud detection research. *Informatica Economica*, *16*(1), 110.

Sáez, J. A., Krawczyk, B., & Woźniak, M. (2016). Analyzing the oversampling of different classes and types of examples in multi-class imbalanced datasets. *Pattern Recognition*, *57*(i), 164–178.

Sahin, Y., Bulkan, S., & Duman, E. (2013). A cost-sensitive decision tree approach for fraud detection. *Expert Systems with Applications*, *40*(15), 5916–5923.

Sánchez, D., Vila, M. A., Cerda, L., & Serrano, J.-M. (2009). Association rules applied to credit card fraud detection. *Expert Systems with Applications*, *36*(2), 3630–3640.

Satyasree, K. P. N. V, & Murthy, J. V. R. (2013). An Exhaustive Literature Review on Class Imbalance Problem. *International Journal of Emerging Trends & Technology in Computer Science*, *2*(3), 109–118.

Seksaria, K. (2016). How do ATM machines work internally? Retrieved August 12, 2017, from https://www.quora.com/How-do-ATM-machines-work-internally

Sharma, N. (2012). Analysis of different vulnerabilities in auto teller machine transactions. *Journal of Global Research in Computer Science*, *3*(3), 38–40.

Strobl, C., & Zeileis, A. (2008). Why and how to use random forest variable importance measures (and how you shouldn't). In *The R User Conference*.

Şengür, E. D. (2010). *İşletmelerde Hile, Hilelerin Önlenmesi, Hileli Finansal Raporlama ve İlgili Düzenlemeler ve Bir Araştırma*. İstanbul University.

Tasoulis, D., Adams, N., Weston, D. J., & Hand, D. J. (2008). Mining information from plastic card transaction streams. In *Proceedings in Computational Statistics: 18th Symposium (COMPSTAT 2008)* (Vol. 2, pp. 315–322).

Van Vlasselaer, V., Bravo, C., Caelen, O., Eliassi-Rad, T., Akoglu, L., Snoeck, M., & Baesens, B. (2015). APATE: A novel approach for automated credit card transaction fraud detection using network-based extensions. *Decision Support Systems*, *75*, 38–48.

Van Vlasselaer, V., Eliassi-Rad, T., Akoglu, L., Snoeck, M., & Baesens, B. (2016). Gotcha! Network-based fraud detection for social security fraud. *Management Science*.

Varıcı, İ. (2011). *Hile Riski ve Denetçinin Sorumluluğu: Hile Riskinin Ölçülmesine Yönelik Bir Uygulama*. Karadeniz Teknik Üniversitesi.

Vona, L. W. (2008). *Fraud Risk Assessment : Building a Fraud Audit Program*. Hoboken, NJ: Wiley.

Wen, G., Jiang, L., & Wen, J. (2014). Multiple perceptual neighborhoods-based feature construction for pattern classification. *Neurocomputing*, *142*, 499–507.

West, J., & Bhattacharya, M. (2016). Intelligent financial fraud detection: A comprehensive review. *Computers and Security*, *57*, 47–66.

Whitrow, C., Hand, D. J., Juszczak, P., Weston, D., & Adams, N. M. (2009). Transaction aggregation as a strategy for credit card fraud detection. *Data Mining and Knowledge Discovery*, *18*(1), 30–55.

Wikipedia. (2017a). ATM Card. Retrieved August 15, 2017, from https://en.wikipedia.org/wiki/ATM_card

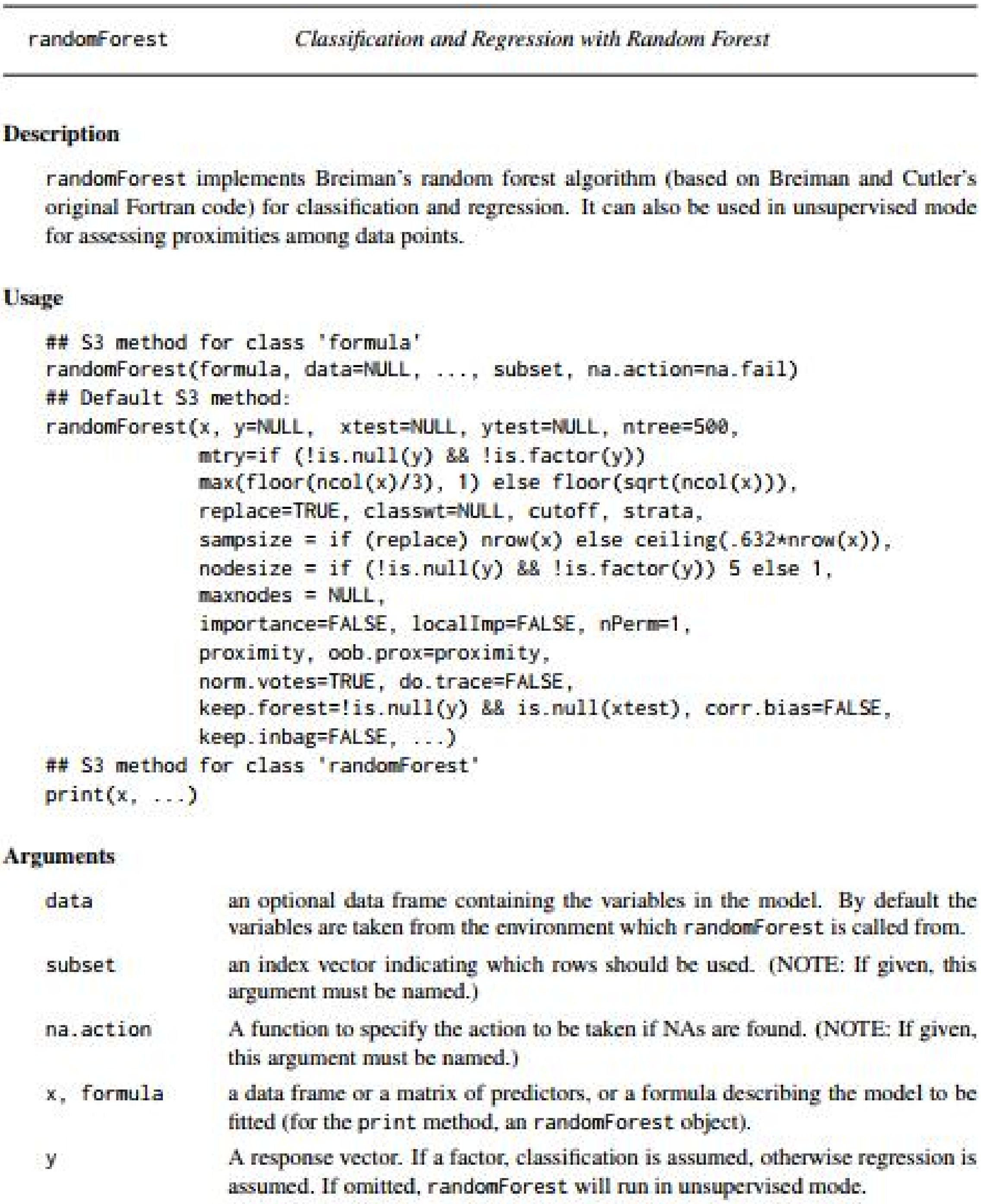
Wikipedia. (2017b). Automated Teller Machine. Retrieved July 29, 2017, from https://en.wikipedia.org/wiki/Automated_teller_machine

Wikipedia. (2017c). Debit Card. Retrieved August 15, 2017, from https://en.wikipedia.org/wiki/Debit_card

Xing, D., & Girolami, M. (2007). Employing Latent Dirichlet Allocation for fraud detection in telecommunications. *Pattern Recognition Letters*, *28*(13), 1727–1734.

APPENDICES

Appendix-1: Manual of RandomForest Formula



`randomForest` *Classification and Regression with Random Forest*

Description

`randomForest` implements Breiman's random forest algorithm (based on Breiman and Cutler's original Fortran code) for classification and regression. It can also be used in unsupervised mode for assessing proximities among data points.

Usage

```
## S3 method for class 'formula'
randomForest(formula, data=NULL, ..., subset, na.action=na.fail)
## Default S3 method:
randomForest(x, y=NULL,  xtest=NULL, ytest=NULL, ntree=500,
             mtry=if (!is.null(y) && !is.factor(y))
             max(floor(ncol(x)/3), 1) else floor(sqrt(ncol(x))),
             replace=TRUE, classwt=NULL, cutoff, strata,
             sampsize = if (replace) nrow(x) else ceiling(.632*nrow(x)),
             nodesize = if (!is.null(y) && !is.factor(y)) 5 else 1,
             maxnodes = NULL,
             importance=FALSE, localImp=FALSE, nPerm=1,
             proximity, oob.prox=proximity,
             norm.votes=TRUE, do.trace=FALSE,
             keep.forest=!is.null(y) && is.null(xtest), corr.bias=FALSE,
             keep.inbag=FALSE, ...)
## S3 method for class 'randomForest'
print(x, ...)
```

Arguments

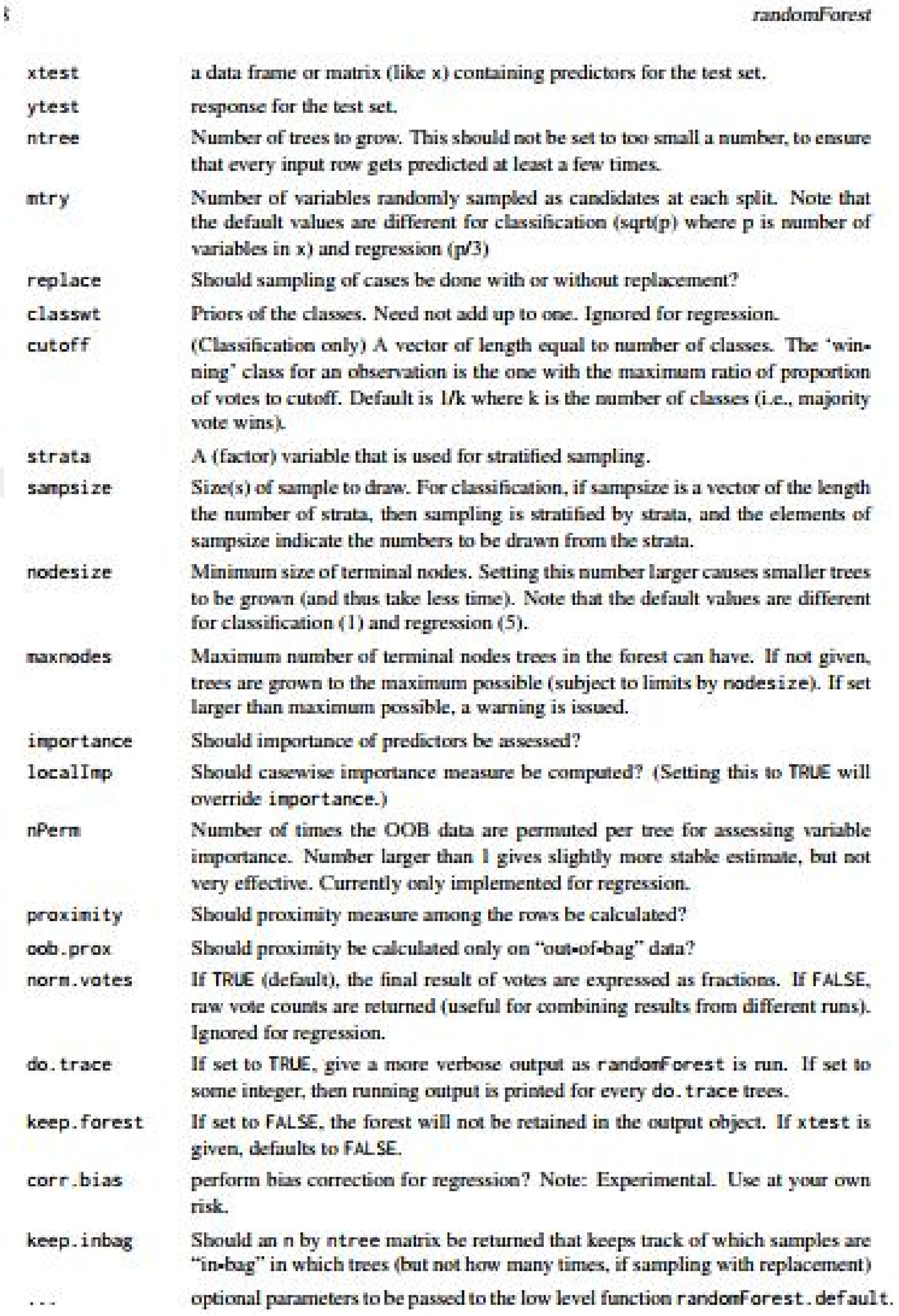
`data` an optional data frame containing the variables in the model. By default the variables are taken from the environment which `randomForest` is called from.

`subset` an index vector indicating which rows should be used. (NOTE: If given, this argument must be named.)

`na.action` A function to specify the action to be taken if NAs are found. (NOTE: If given, this argument must be named.)

`x, formula` a data frame or a matrix of predictors, or a formula describing the model to be fitted (for the `print` method, an `randomForest` object).

`y` A response vector. If a factor, classification is assumed, otherwise regression is assumed. If omitted, `randomForest` will run in unsupervised mode.



`xtest` a data frame or matrix (like x) containing predictors for the test set.

`ytest` response for the test set.

`ntree` Number of trees to grow. This should not be set to too small a number, to ensure that every input row gets predicted at least a few times.

`mtry` Number of variables randomly sampled as candidates at each split. Note that the default values are different for classification (sqrt(p) where p is number of variables in x) and regression (p/3)

`replace` Should sampling of cases be done with or without replacement?

`classwt` Priors of the classes. Need not add up to one. Ignored for regression.

`cutoff` (Classification only) A vector of length equal to number of classes. The 'winning' class for an observation is the one with the maximum ratio of proportion of votes to cutoff. Default is 1/k where k is the number of classes (i.e., majority vote wins).

`strata` A (factor) variable that is used for stratified sampling.

`sampsize` Size(s) of sample to draw. For classification, if sampsize is a vector of the length the number of strata, then sampling is stratified by strata, and the elements of sampsize indicate the numbers to be drawn from the strata.

`nodesize` Minimum size of terminal nodes. Setting this number larger causes smaller trees to be grown (and thus take less time). Note that the default values are different for classification (1) and regression (5).

`maxnodes` Maximum number of terminal nodes trees in the forest can have. If not given, trees are grown to the maximum possible (subject to limits by `nodesize`). If set larger than maximum possible, a warning is issued.

`importance` Should importance of predictors be assessed?

`localImp` Should casewise importance measure be computed? (Setting this to `TRUE` will override `importance`.)

`nPerm` Number of times the OOB data are permuted per tree for assessing variable importance. Number larger than 1 gives slightly more stable estimate, but not very effective. Currently only implemented for regression.

`proximity` Should proximity measure among the rows be calculated?

`oob.prox` Should proximity be calculated only on "out-of-bag" data?

`norm.votes` If `TRUE` (default), the final result of votes are expressed as fractions. If `FALSE`, raw vote counts are returned (useful for combining results from different runs). Ignored for regression.

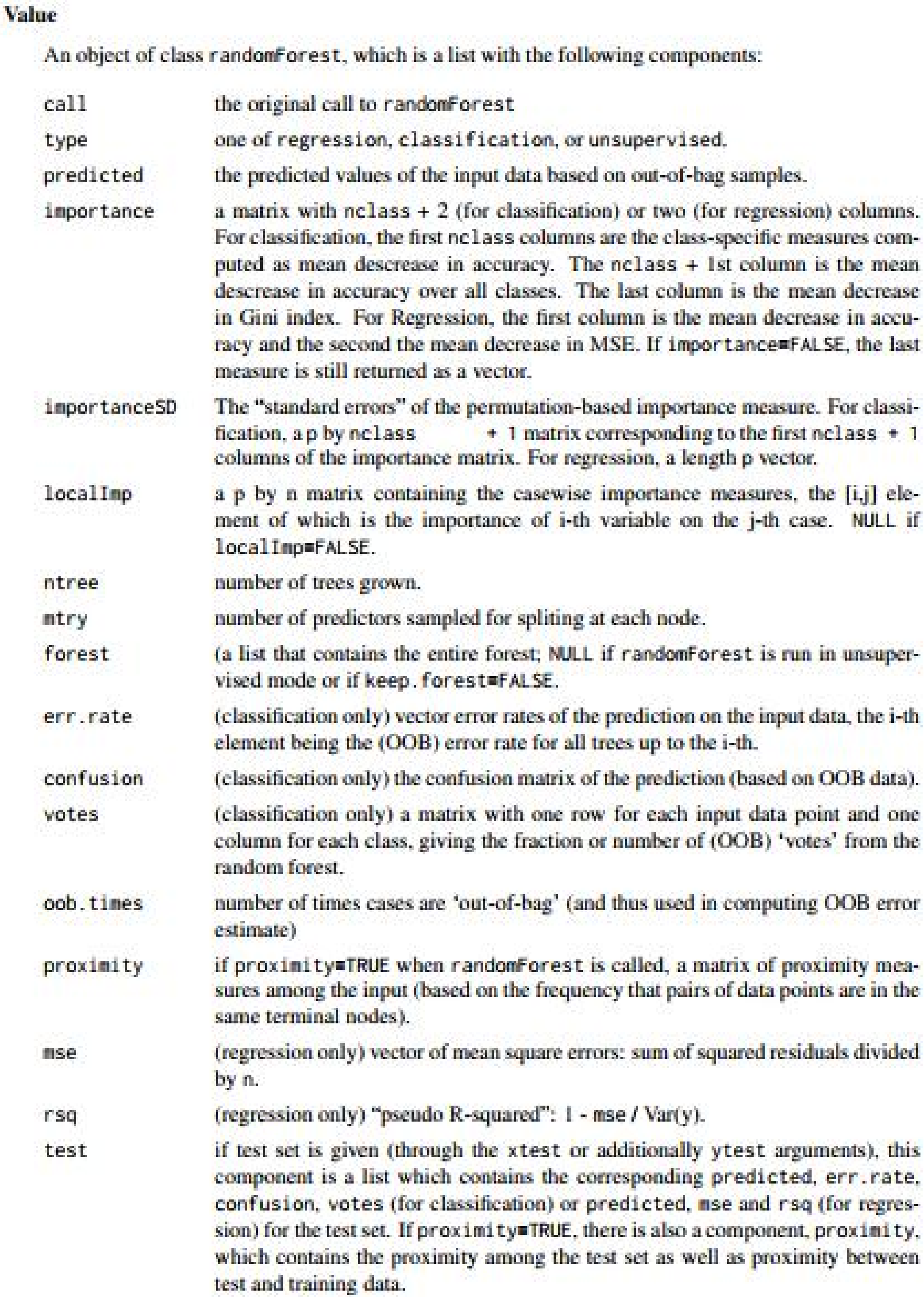
`do.trace` If set to `TRUE`, give a more verbose output as `randomForest` is run. If set to some integer, then running output is printed for every `do.trace` trees.

`keep.forest` If set to `FALSE`, the forest will not be retained in the output object. If `xtest` is given, defaults to `FALSE`.

`corr.bias` perform bias correction for regression? Note: Experimental. Use at your own risk.

`keep.inbag` Should an `n` by `ntree` matrix be returned that keeps track of which samples are "in-bag" in which trees (but not how many times, if sampling with replacement)

`...` optional parameters to be passed to the low level function `randomForest.default`.



Value

An object of class `randomForest`, which is a list with the following components:

`call` the original call to `randomForest`

`type` one of `regression`, `classification`, or `unsupervised`.

`predicted` the predicted values of the input data based on out-of-bag samples.

`importance` a matrix with `nclass` + 2 (for classification) or two (for regression) columns. For classification, the first `nclass` columns are the class-specific measures computed as mean descrease in accuracy. The `nclass` + 1st column is the mean descrease in accuracy over all classes. The last column is the mean decrease in Gini index. For Regression, the first column is the mean decrease in accuracy and the second the mean decrease in MSE. If `importance=FALSE`, the last measure is still returned as a vector.

`importanceSD` The "standard errors" of the permutation-based importance measure. For classification, a `p` by `nclass + 1` matrix corresponding to the first `nclass + 1` columns of the importance matrix. For regression, a length `p` vector.

`localImp` a p by n matrix containing the casewise importance measures, the [i,j] element of which is the importance of i-th variable on the j-th case. `NULL` if `localImp=FALSE`.

`ntree` number of trees grown.

`mtry` number of predictors sampled for spliting at each node.

`forest` (a list that contains the entire forest; `NULL` if `randomForest` is run in unsupervised mode or if `keep.forest=FALSE`.

`err.rate` (classification only) vector error rates of the prediction on the input data, the i-th element being the (OOB) error rate for all trees up to the i-th.

`confusion` (classification only) the confusion matrix of the prediction (based on OOB data).

`votes` (classification only) a matrix with one row for each input data point and one column for each class, giving the fraction or number of (OOB) 'votes' from the random forest.

`oob.times` number of times cases are 'out-of-bag' (and thus used in computing OOB error estimate)

`proximity` if `proximity=TRUE` when `randomForest` is called, a matrix of proximity measures among the input (based on the frequency that pairs of data points are in the same terminal nodes).

`mse` (regression only) vector of mean square errors: sum of squared residuals divided by `n`.

`rsq` (regression only) "pseudo R-squared": 1 - `mse` / Var(y).

`test` if test set is given (through the `xtest` or additionally `ytest` arguments), this component is a list which contains the corresponding `predicted`, `err.rate`, `confusion`, `votes` (for classification) or `predicted`, `mse` and `rsq` (for regression) for the test set. If `proximity=TRUE`, there is also a component, `proximity`, which contains the proximity among the test set as well as proximity between test and training data.

Appendix-2: Importance of Variables in the Base Scenario

Variable Name	Mean Decrease Accuracy*1000	Mean Decrease Gini
TimeSince_All_3Month	5.79464	17.90502217
TimeSince_Currency_3Month	5.54834	14.96437008
RatioAverage_Currency_3Month	5.0566	1.81366411
Amount_Group_3Month	4.96636	1.78649616
RatioAverage_All_3Month	4.7259	1.54155567
Amount_Currency_3Month	4.5779	2.25676622
Amount_All_3Month	4.28476	2.81852399
Average_Currency_3Month	4.1595	1.01829954
TimeSince_Group_3Month	4.03566	11.78857246
Average_All_3Month	3.99333	1.10807356
RatioAverage_ATM_3Month	3.82448	2.9471434
RatioAverage_Group_3Month	3.54941	1.37060904
Count_Group_3Month	3.49334	10.22941294
DiffAverage_Currency_3Month	3.47625	1.80923618
DiffAverage_Group_3Month	3.40914	1.82819807
DiffAverage_All_3Month	3.2772	1.77286281
DiffAverage_All_Month	3.2198	1.42116581
RatioAverage_ATM_Month	3.17948	2.06284628
RatioAverage_All_Month	3.07173	1.20261991
TimeSince_Currency_Month	3.05857	17.92814071
Count_All_3Month	3.02563	26.81433089
DiffAverage_ATM_3Month	2.9512	2.68390375
Count_Currency_3Month	2.94339	19.92608489
PreviousBalance	2.8265	1.89909058
TimeSince_All_Month	2.75613	17.15173193
DiffAverage_Currency_Month	2.69286	1.42662178
RatioAverage_Currency_Month	2.63764	1.29572251
RatioAverage_Group_Month	2.61626	1.02681174
RatioAverage_Currency_Week	2.57572	1.31776609
RatioAverage_All_Week	2.53162	1.35211965
DiffAverage_Group_Month	2.49213	1.29886878
TimeSince_Group_Month	2.41616	12.54153293
DiffAverage_ATM_Month	2.36032	2.14522276
Average_Group_3Month	2.13721	0.95463831
DiffAverage_Group_Week	1.9751	1.00814437
RatioAverage_Group_Week	1.91402	1.25170431
Amount_Group_Month	1.88768	1.42308958
RatioAverage_ATM_Week	1.8411	1.39597991
DiffAverage_Currency_Week	1.78084	0.99039172
DiffAverage_All_Week	1.76684	0.89161548
DiffAverage_ATM_Week	1.69675	1.53856723
DiffAverage_Group_Hour	1.67961	0.88408797

Count_Group_Month	1.66386	11.97026148
DiffAverage_Group_Day	1.58856	0.93055259
Amount_All_Month	1.58299	2.2689492
Average_All_Month	1.57162	0.94422427
DiffAverage_ATM_Day	1.56108	0.97631212
RatioAverage_All_Hour	1.53594	0.97918381
RatioAverage_ATM_Day	1.53132	1.3550317
Count_All_Month	1.49961	30.04408546
TimeSince_Currency_Week	1.43761	15.86725706
DiffAverage_All_Hour	1.42019	0.91555663
DiffAverage_ATM_Hour	1.41142	1.0032673
DiffAverage_All_Day	1.39104	0.86033571
Count_Currency_Month	1.3831	23.23901579
RatioAverage_ATM_Hour	1.37581	1.13996736
RatioAverage_Group_Hour	1.3754	1.08890197
Amount_Currency_Month	1.35881	1.10036856
RatioAverage_Currency_Day	1.3561	0.95395309
TimeSince_All_Week	1.3259	17.33776112
RatioAverage_Group_Day	1.29276	1.28591163
RatioAverage_All_Day	1.27671	0.91084768
Amount_Currency_Week	1.25113	7.66152058
Amount_All_Week	1.24415	9.14444119
RatioAverage_Currency_Hour	1.23346	0.97229704
AmountinTL	1.20962	1.15134708
Average_All_Week	1.19258	7.25610857
DiffAverage_Currency_Hour	1.18885	0.9486988
Average_Group_Month	1.17339	0.65038459
LargestTransaction	1.15472	1.56813428
WithdrawalRatio	1.13874	1.92064403
Average_Currency_Week	1.08743	6.48682869
Average_Currency_Month	1.04136	0.76879576
DiffAverage_Currency_Day	1.00847	0.78026833
TimeSince_Group_Week	1.00012	10.18974253
Amount_ATM_3Month	0.98554	1.27065037
Average_Group_Week	0.95627	5.65444429
TimeSince_ATM_3Month	0.84584	0.81783603
Amount_Group_Week	0.80768	5.19732359
Average_ATM_3Month	0.76564	0.9774591
Count_ATM_3Month	0.72797	1.12527436
TimeSinceLastTransaction	0.54668	6.81444569
TimeSince_ATM_Month	0.54524	0.50586934
Age	0.52245	1.72428151
FirstWd_Group_Week	0.52064	3.97190925

FirstWd_Currency_3Month	0.46137	0.06228018
FirstWd_Currency_Week	0.4498	4.0510412
Count_Group_Week	0.42477	3.93665306
TimeSince_ATM_Week	0.4056	0.42041251
FirstWd_All_Week	0.40269	4.30592635
Count_Currency_Week	0.36843	13.58736034
Amount_ATM_Week	0.36662	0.34587109
FirstWd_ATM_3Month	0.3545	0.38438939
Count_All_Week	0.34886	13.61656326
Amount_ATM_Month	0.34212	0.63140487
FirstWd_All_3Month	0.29337	0.04587218
Average_ATM_Week	0.29084	0.35902013
Average_ATM_Month	0.28626	0.59245686
FirstWd_All_Month	0.26015	0.15704755
Count_ATM_Month	0.2235	0.3895229
FirstWd_Currency_Month	0.22316	0.16079931
FirstWd_Group_3Month	0.20855	0.08568277
Education Level	0.1636	1.02223397
FirstWd_Group_Month	0.14876	0.19284445
FirstWd_ATM_Month	0.13956	0.17157087
FirstWd_ATM_Week	0.08309	0.08000072
Count_Group_Day	0.06775	2.05897412
Count_ATM_Week	0.06656	0.14487437
Average_ATM_Day	0.04979	0.13818875
FirstWd_All_Day	0.04664	1.93076073
Average_All_Day	0.04454	1.43158835
Average_Group_Day	0.03697	1.1414078
Count_Currency_Day	0.03261	1.00019162
Amount_Group_Day	0.02798	1.06305501
FirstWd_Currency_Day	0.02477	1.64849603
Amount_ATM_Day	0.02208	0.12805888
FirstWd_Group_Day	0.02115	1.18853849
Count_All_Day	0.02059	1.99108199
TimeSince_Group_Day	0.01771	1.11573807
TimeSince_Currency_Day	0.01758	3.24246757
Average_ATM_Hour	0.01655	0.10486039
FirstWd_ATM_Day	0.01441	0.02058309
FirstWd_Currency_Hour	0.0131	0.06100242
Average_Currency_Day	0.01154	1.60318957
Count_ATM_Hour	0.01026	0.03106005
TimeSince_ATM_Day	0.00596	0.11164922
Amount_ATM_Hour	0.00575	0.05482928
Count_ATM_Day	0.00515	0.02824207

Count_Currency_Hour	0.005	0.07601693
Average_Group_Hour	0.00442	0.087253
Amount_Currency_Day	0.00309	1.38508371
Currency	0.00288	0.17751275
Count_All_Hour	0.00091	0.0844992
FirstWd_ATM_Hour	0.00027	0.01059365
Amount_Group_Hour	-0.00044	0.06493854
Average_Currency_Hour	-0.00087	0.07683966
Amount_All_Hour	-0.001	0.26908847
FirstWd_Group_Hour	-0.0017	0.01307552
FirstWd_All_Hour	-0.00208	0.12413763
TimeSince_ATM_Hour	-0.00297	0.05653086
Amount_Currency_Hour	-0.00374	0.12205249
TimeSince_Group_Hour	-0.00392	0.10608721
Count_Group_Hour	-0.00461	0.0357783
TimeSince_Currency_Hour	-0.00495	0.14591132
Average_All_Hour	-0.00639	0.19450002
TimeSince_All_Hour	-0.00736	0.27496237
IsOurBank	-0.0102	0.08005873
Amount_All_Day	-0.01557	2.51782736
TimeSince_All_Day	-0.05642	3.6857698