University of Coimbra

Master in Informatics Engineering

Internship in Software Engineering

Alert Manager App for e-Commerce Customers

Final Report

Maria Inês Oliveira Pinto Coelho
micoelho@student.dei.uc.pt

Advisors:
Penousal Machado
António Alegria (Feedzai)
Rafael Marmelo (Feedzai)

Jury:
Bruno Cabral
Ernesto Costa

August 31, 2016

To my parents, who put up with me through thick and thin.

Abstract

In the electronic world we live in, where more and more commercial transactions happen over the internet and where the magnitude of losses to fraud increases every day, the need for sophisticated systems for fraud detection and prevention arise.

One of the pitfalls of existing system is the inability to make sense of the extensive amount of business data available. Significant insights could be obtained by identifying relationships among that data.

Entity Link analysis emerge as a powerful ally of fraud detection and prevention systems. By exploring relationships within a network of related entities, abnormal patterns of behaviour can be uncovered.

In this report we describe the development of an exploratory tool for Entity Linking. This web application provides different data visualizations to analyse commercial transactions and identify relationships between them, in order to detect fraudulent entities. Data visualizations available are: table, matrix diagram, geographical referencing, force directed graph, circular diagram and chord diagram.

Through usability evaluation, force directed graph was elected as the most promising visualization for fraud analysis. But both the matrix diagram and the circular diagram obtained good reviews, opening the discussion for a combined use of data visualizations to be integrated with the fraud detection and prevention solution.

KeyWords: Alert Manager, Fraud, Electronic Commerce, Link Analysis, Entity Linking, Data Visualizations

Acknowledgments

First, and foremost, I want to thank my supervisors, Rafael Marmelo, António Alegria and Penousal Machado. Without their assistance and dedication, this dissertation would have never been accomplished. I would like to thank you all for your support and understanding, and for steering me in the right direction, whenever I needed it.

I would also like to thank Feedzai, for granting me this opportunity and for welcome me as an intern with open arms. Without their precious support, it would have not been possible to develop this project. A special thank you note goes to my co-workers for all the feedback granted and for voluntarily participating on this project's usability evaluation.

But, this thesis is the culmination of an academic experience, and I would like to use this opportunity to express my gratitude to everyone who supported me throughout it. I am thankful for all the people I've met along the way and with I had the pleasure to work it. I am sincerely grateful to have shared this experience along you and to have grown by your side. A special thanks to Tiago, João A., João Tiago and Sandra for all the support and guidance.

My friends have always been a cornerstone in my life. First, I want to thank Folharascas, for being in my life and put up with me for so long, and the Gang, who have always been there for me. A thank you note also needs to be addressed to Isabel and Barbosa, my life wouldn't have been the same without all of our happy mess. A special thanks goes to Mariana N., for all the friendship and the long coffee breaks at odd hours special, Daniela and Filipa, who always know how to put a smile on my face, and Carlos and Daniel who supported me through all of this. I could add many more names to this list, but... you know who you are, and how special you are for me.

Last, but certainly not the least, I want to address a huge thank you to my family, especially my parents and my sister. You are the reason of what I have become today and this accomplishment would not have been possible without you. This dissertation stands as a testimony of your unconditional love and encouragement. Thank you, from the bottom of my heart.

And now for something completely different.
Monty Python

Contents

List of Tables

List of Figures

Acronyms

AM: Alert Manager

AMS: Alert Management System

API: application programming interface

CNP: Card-Not-Present

CP: Card-Present

DOM: Document Object Model

e-Commerce: Electronic Commerce

HTTP: Hypertext Transfer Protocol

IT: Information Technology

JAX-RS: Java API for RESTful Web Services

JDBC: Java Database Connectivity

MVP: Model-View-Presenter

POJO: Plain Old Java Object

RDBMS: Relational Database Management System

REST: Representational State Transfer

SaaS: Software as a Service

SVG: Scalable Vector Graphics

SUS: System Usability Scale

UI: User Interface

1 Introduction

The intent of this introductory chapter is to contextualize this Master Thesis, clarifying its scope, goals and motivation, and to present the structure of this report.

1.1 Purpose

The present document describes the design, implementation and testing of data visualizations for Entity Link Analysis in fraud detection. These visualizations are aimed to be integrated in the Alert Manager application, a component from a fraud prevention solution for online commerce, being currently developed by Feedzai.

This project was developed in the scope of the Dissertation/Internship in Software Engineering for the Master's Degree in Informatics Engineering at the University of Coimbra, during the academic year 2015/2016.

1.2 Motivation

With the rise of the Internet, a revolution in commerce occurred. Business activities extended their reach to the electronic world and, with it, criminal activities followed behind. Every year, banks, insurance companies and commerce activities lose significant amounts of money to electronic fraud [52].

Furthermore, as technology evolves, fraudsters become more sophisticated. They use elaborated scams to construct false identities and, sometimes, they work together to establish fraud rings, which are even more complex to uncover.

Traditional methods of fraud detection are insufficient to fight this type of crimes. Gathering new data is not always the solution of the problem; sometimes it is necessary to reframe the problem and look at it in another perspective, to make sense of the data already collected and draw significant insights. It is crucial to look beyond individual data points to find the connections among them, draw out hidden patterns and to expose the picture concealed beneath.

Link analysis is a technique used to identify relationships among data objects. This knowledge discovery process focuses in analysing connections among node objects through data visualization methods. Data visualizations generated automatically by link-analysis tools can uncover patterns that are difficult to detect using traditional representations, such as tables. When applied to financial transactions, it can be used to investigate criminal activities and uncover connections between fraudster entities.

Therefore, those data visualizations might be a powerful tool to disclose relationships among fraudsters, identify entities committing fraud, uncover fraud rings and hinder the advance of digital scams in real time.

1.3 Context

The internship took place at Feedzai, a data science company specialized in fraud detection and prevention. Taking advantage of machine learning to detect fraud, the technology developed at Feedzai aims to understand the way customers behave when they make purchases both online and in-store.

Their main product, Feedzai Pulse, is a Fraud Prevention solution that employs next generation big data analytics to deliver the industry's best millisecond-latency fraud blocking capabilities, with an extremely low number of false positives. Its machine learning capability instantly identifies new fraudster schemes, stopping fraudulent transactions on their first occurrence.

Pulse high-level functioning is depicted on figure 1, with a burgundy colouring. When a customer performs a transaction (e.g. ATM payment, purchase in a shop (merchant's POS or online – identified on the picture as 1), the network automatically asks Feedzai Pulse to analyse it (2). Pulse scores the transaction, storing the result into a datastore (3), and returns the result to the Network (4), which will accept or block the transaction and communicate with the bank (5 and 6) and the transaction terminal (7).

However, there is a certain risk threshold in which blocking transactions automatically can hassle reliable customers, affecting the business from service providers. In those cases, manual review of the transaction by humans might be the best course of action. The new Alert Manager Application, currently under development, will focus on this domain. When a transaction cannot be automatically processed, an alert is generated to notify the operators of the need to manually review the transaction.

Flow of information on Alert Manager is also portrayed on figure 1, with a green colouring. When an alert is emitted, Alert Manager will enrich the transaction in the datastore with relevant contextual information (1) and display it to a human analyst for review (2). Resorting to the data visualizations presented, the analyst marks the transaction as fraudulent or legit (3) and this information is returned to Pulse (4) to reinforce Pulse detection rules.

Therefore, the target users of Alert Manager are fraud analysts that manage risk internally, in a company, and need a platform to analyse and to review alerts. This application provides context and information about the transaction that generated the alert, helping fraud analysts to decide the best outcome for each alert. The developed data visualizations for Entity Link Analysis will be crucial to this process.

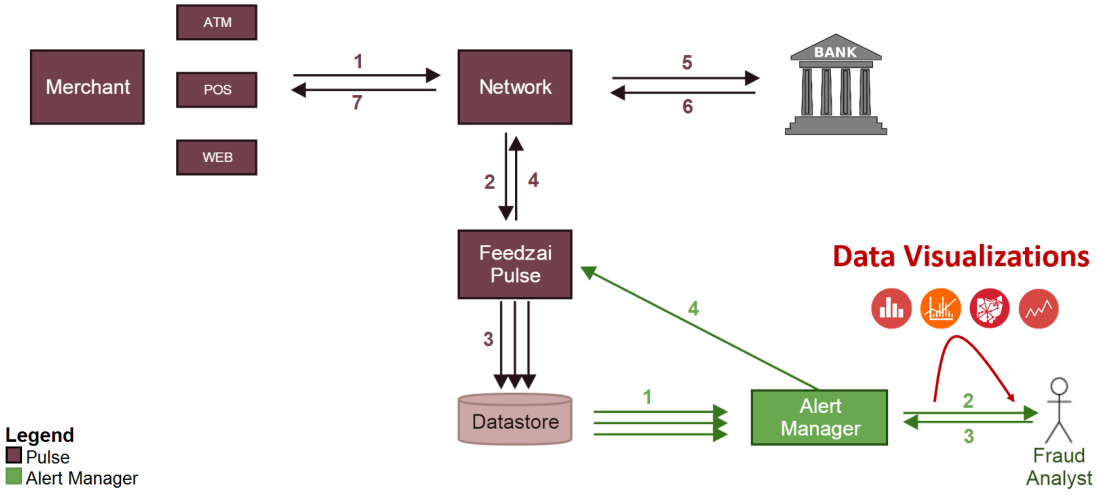

Figure 1: High-level view of a network-wide deployment featuring Feedzai Pulse and Alert Manager.

1.4 Scope

The goal of the Alert Manager application is to provide a User Interface for fraud analysts quickly and efficiently distinguish fraud cases from legitimate cases. To that end, it's necessary to identify patterns, establish relationships and detect outliers on commercial transaction's enriched data. The results must be presented through a powerful and innovative dashboard, supported by rich visualizations, to help operators search, navigate and correlate data, using an analytic approach, so they can make the best possible informed decision.

This internship project consists on the design, development and testing of data visualizations for the Alert Manager application in order to identify entities behind transactions and existing relationships between them.

1.5 Objectives

Work on this project was integrated with the development team currently working in the Alert Manager Application at Feedzai.

There were several kinds of information, datapoints and patterns that could be explored in the scope of this application. After a thoroughly analysis, we choose to focus solely on the subset where a bigger impact can be gained by doing innovative visualizations: the identification of entities behind fraud in e-commerce transactions. Therefore, the goal of this project is to propose, implement and test a set of innovative data visualizations for Entity Link Analysis for fraud detection and select the most appropriate visualization to be integrated within the Alert Manager Application.

From an academic point of view, the goals of this project are to consolidate knowl-

edge in Software Engineering and in web development technologies, to gain experience while developing software in a corporate environment, with real life clients and all its implications, and to learn from this experience.

1.6 Contributions

In particular, the following contributions were made:

- **State of the Art Review:** detailed overview of the background concepts required to support this work; structured analysis of different visualization types and where it makes sense applying them or not; comparative analysis on related systems.
- **Exploratory Tool for Entity Link Analysis:** web application featuring a set of interactive data visualizations for Entity Link Analysis, supporting business data from different sources.
- **Usability Evaluation:** system evaluation and list of recommendations to improve the usability of the system and the efficiency of the data visualizations

1.7 Overview

This report is structured in eleven chapters, organized as follows:

1. Introduction: overview of the internship project, including purpose, motivation, context, scope, objectives and contributions.
2. Background Knowledge: presentation of concepts related with this project, such as fraud, fraud detection and prevention, entity link analysis and data visualizations.
3. State of the Art: comparative analysis of Fraud Detection Systems for e-commerce, with Alert Management capabilities.
4. Work Plan: description of the development methodologies used and work planning.
5. Requirements Specification: specification of the project features, quality attributes and constraints.
6. Architecture: presentation of the application architecture and an overview of the technical decisions made during project development.
7. Risk Management: identification and management of potential problems that could undermine this project.
8. Implementation: detailed look at the main components of the Exploratory Tool built.

9. Validation: description of the tests used to validate the exploratory data visualizations and confirm that the defined requirement were met, and list of resulting recommendations to improve the system.
10. Future Work: discussion and exploration of how the system can be improved.
11. Conclusion: review and discussion of the project.

2 Background Knowledge

This introductory chapter sets the scene for this thesis. It starts by defining and characterizing fraud in electronic commerce, presenting different types of fraud and how fraudsters work. Afterwards, fraud detection and prevention is discussed. Entity link analysis and data visualizations are featured as powerful tools that may improve fraud detection systems.

2.1 Fraud in Electronic Commerce

A crime in which some kind of deception is used for personal gain is designated as fraud [21, 36]. In the United States of America, federal law defines electronic fraud as "the use of a computer to create a dishonest misrepresentation of certain facts, in an attempt to induce other person to do or to refrain from doing something with detrimental consequences for that person" [35].

Electronic fraud has become more sophisticated as technology evolves. Among the main types of electronic fraud are [67]:

- Identity Theft - criminals access the victim personal information and steal their identity, using their name and reputation for their own financial gain;
- Credit Card Fraud - either by card skimming, cloning techniques or by stealing, fraudsters take charge of the victim's credit card and extort or spend huge amounts of money;
- Phishing - through fake emails and web links, fraudster obtain sensitive information about their victims, such as passwords, usernames or bank account details.

Electronic commerce comprises business transactions over an electronic network, such as the internet. Every day, e-commerce is getting more and more attention. In 2013, it was estimated that 21-40 % of a business contributions were a direct result of e-commerce and this number keeps on growing [7]. Fraud in e-commerce occurs when a business transaction is performed, but the criminal uses fraudulent means to pay it. In 2012, fraud rate in e-commerce ranged, on average, 0.9 % [29].

2.1.1 Types of Fraud in Electronic Commerce

In e-commerce, interactions are usually of the card-not-present type. The merchant has to make instant decisions, becoming an easy prey to fraudsters.

The two most common types of fraud that impact e-commerce are [44]:

- Chargeback Fraud: after an online purchase is made using a credit card, a chargeback request can be presented by the credit card owner. In that case, the merchant risks to lose both the money from the transaction and the merchandise, if already shipped, as result. This kind of fraud can be carried out

using false or stolen credit cards. However, in some cases, the order may actually have been placed by someone close to the account holder, such as a family member or a friend, who has access to the account information.

- Account Takeover Fraud: either by stealing information, exploring data breaches or other means, fraudsters gain access to an account's personal information. Using those stolen credentials, the identity thieves can cut the access by the respective owner, taking over the account and pretending to be the victim. They use the account to make as many purchases as possible until they are detected and their access to the account is withdrawn.

For many years, credit cards have been consistently the largest source of payment fraud. In the United States, it was estimated that, in 2014, credit cards contributed to 52 % of all fraudulent payments and that, in 2013, one out of every seven payment cards used was exposed in a data breach [31].

Credit card fraud affects more than the people directly involved in the transaction. Besides the customer, whose credit card information was stolen, and the merchant, whose product was purchased, credit card frauds also involve: the bank that facilitates the transaction; the issuer who is charged with protecting its cardholders; the payment company who needs to invest a lot of money in fraud prevention measures [92]. Often, in order to avoid chargebacks, merchants opt to refund their customers [81].

On the other hand, identity theft is one of the fastest growing types of fraud [76]. Services on the internet often lack the client-side security needed to protect personal information from their clients, entrusted to their care. They are too complacent to identity theft attacks, when protecting their customers' databases to the fullest extent possible should be their duty and ethical responsibility. E-commerce sites should have a privacy policy, adequate encryption measures, suitable site security and should carefully check each transaction for fraud [86].

2.1.2 Internet Fraudsters

Hand in hand with technological development, internet fraudsters evolve and adapt. They are the entities behind the crimes, relying upon layers of indirection to deceive merchants and to cover their tracks.

Sophisticated fraudsters prime by the variety of scams they develop to elude discovery. They disguise themselves, creating multiple identities to commit fraud [69]. Their deception is achieved by switching and combining different credit cards, emails, devices, locations, phone numbers, etc., therefore creating a huge number of synthetic identities that they use to carry on their fraudulent schemes.

This scenario is difficult to deal with, because variations between transactions are natural to occur. An individual can use multiple computers or devices to perform transactions and, likewise, different individuals can share the same device. A credit card number can be personal or shared by a family. Similarly, multiple credit cards can have the same address. Often, a transaction billing address is the same as the

shipping address; other times, the same credit card can be used to ship a large number of items to different addresses. However, when the relationship between different parameters exceed a reasonable number, fraud might be at play and, often, fraudulent schemes exhibit different patterns than those previously described (figure 2).

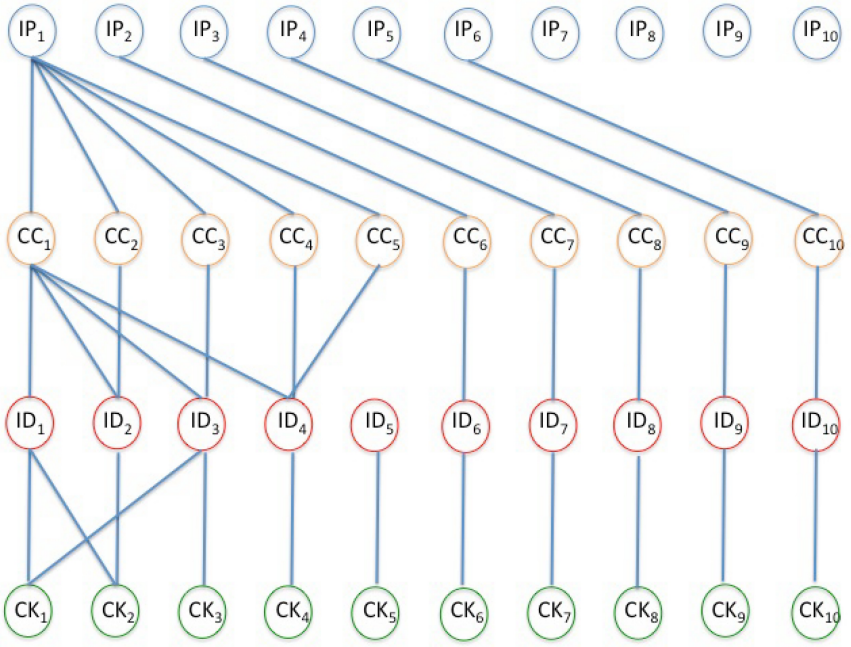

Figure 2: Creation of synthetic identities by combining different identifiers. This graph represent a series of online transaction originated from different IP addresses. Each transaction is characterized by the following identifiers: user ID (ID), IP address (IP), tracking cookie (CK) and credit card number (CC). A fraud event is likely to have occurred from the first IP (IP1), which carried out multiple transactions using five different credit cards, one of which (CC1) is used by multiple IDS, where two cookies (CK1 and CK2) share two IDs each [75].

Internet fraudsters can collaborate together to commit fraud, forming elaborated fraud rings. In 2012, it was estimated that more than 10.000 identity fraud rings were active in the USA [25]. These rings act by controlling multiple identities and their power grows exponentially: a two people ring can control four identities (figure 3), while three people can control nine. Adding a fourth person to the ring expands the number to sixteen, and so on. Furthermore, each identity can control several bank accounts, making these schemes potentially very damaging [75].

2.2 Fraud Detection and Prevention in Electronic Commerce

More and more, merchants are becoming aware of the dangers posed by electronic fraud and are beginning to educate themselves to address it. Analysis show that the three merchant categories most vulnerable to web-related fraud are: retailers selling digital goods, international merchants and omnichannel merchants [70].

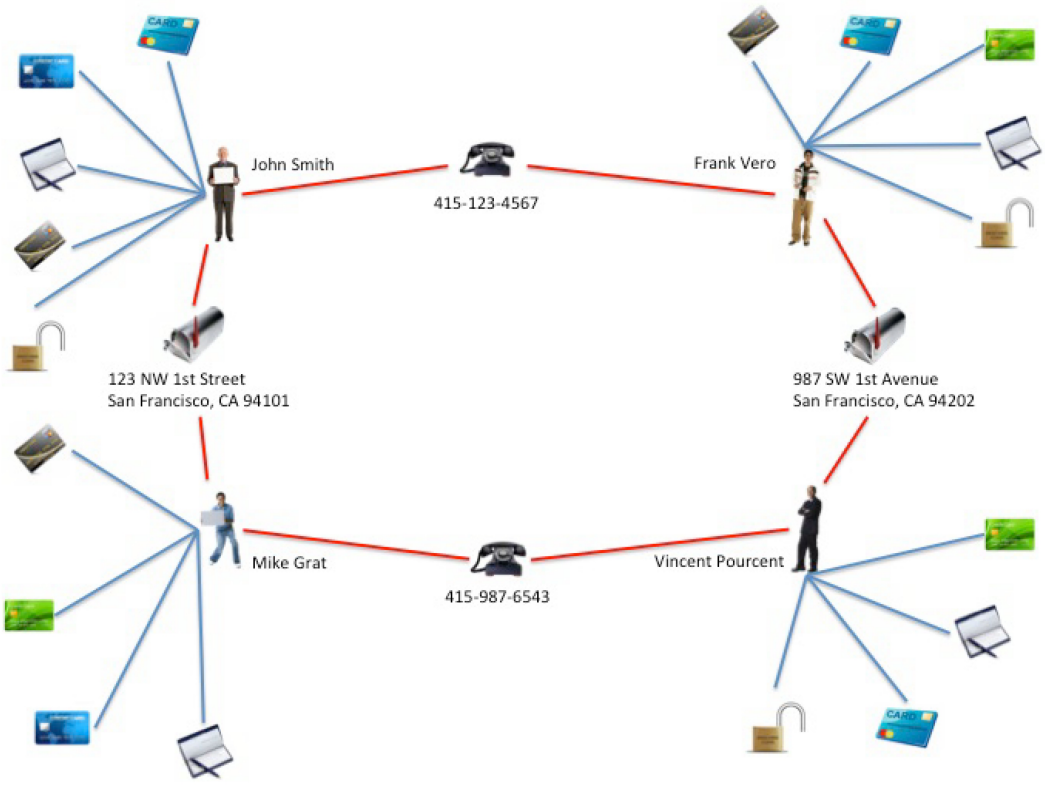

Figure 3: Representation of a fraud ring with two elements.
Two people sharing two pieces of data (phone number and physical address) created four synthetic identities. Each of these identities was used to open several accounts (unsecured credit lines, credit cards, overdraft protection, personal loans, etc.), totalling 18 bank accounts [75].

Proper fraud screening is a fundamental step in any business. Nonetheless, the number of merchants who disregard any form of fraud screening, therefore exposing themselves to fraudsters, is still considerable [81].

Among the merchants that worry with the repercussions of electronic fraud on their business, a significant number still rely, nonetheless, on manual review methods. Manual review is a costly method that requires expertise on the domain in order to identify fraud. Furthermore, it is subjected to human errors and to the occurrence of false positives. However, even using automated systems, fraud mitigation still requires an excessively amount of manual processing. It is estimated that three quarters of the transactions flagged as fraudulent by fraud detection systems still require human intervention to achieve a final decision [50].

Each day, as technology and fraudsters evolve, traditional fraud screening methods become more obsolete. It becomes necessary to search for more effective ways to protect merchants from fraud. There is a serious need to automate fraud screening processes.

Dealing with fraud in e-commerce requires a delicate balance by the merchants: strict rules can drive away potential customers, who will look into the competition

for a less complicated shopping experience; while loose fraud control can lead to credit card chargebacks. Therefore, merchants suffer from fraud losses due to their lack of experience in detecting bad orders. However, turning away or frustrating legitimate users can have a ten times greater impact in the business than the actual fraud losses [83].

2.2.1 Suspicious Orders

One way merchants have of preventing fraud in their business is by monitoring orders, to detect suspicious activities. An attentive merchant knows what his normal orders look like, who are his customers and what are their buying patterns. If anything seems suspicious, it probably is.

Some of the warning signs are [66, 92]:

- Unusual large orders;
- Require alteration of shipping address after the order has been paid;
- Large number of orders at an unusual time;
- Multiple orders from different customers shipped to the same address;
- Suspicious and fake email addresses;
- Suspicious shipping address;
- Rush orders and overnight shipping.

2.2.2 Challenges in Electronic Commerce

The constant technological evolution brings, each day, new challenges to e-commerce merchants. Nowadays, the major challenges faced by merchants, while trying to identify and prevent electronic fraud, are [96]:

- Multitude of communication channels;
- Proliferation of payment types (e.g. bitcoins, e-wallets, SMS payments);
- Marketing expansion.

Moreover, new shopping trends arise, such as business models with same-day/same-hour shipping or digital goods which must be provided immediately. In those cases, merchants have almost no time to properly review the order before accepting it. Hence, the need of a good Alert Management system, to help them make the best informed decision in the less amount of time possible.

Studies show that 50 percent of online retailers have difficulties on keeping up with the latest fraud methods and trends, and 77 percent of the merchants stated that a multichannel payment approach makes fraud more difficult to identify, manage and

prevent [96]. The majority of e-commerce merchants are unaware of techniques to discover and prevent cases of fraudulent payments.

Fraudulent transactions are hard to detect and e-commerce business owners and managers have many other concerns to worry about on their business. Merchants should focus on what they do best, their sales, not in detecting fraud. Since many of e-commerce merchants lack the time, expertise or budget to identify and prevent fraud in this layered environment, they could most certainly benefit from partnering with an expert fraud provider company, which would provide a fraud prevention system tailored to their specific needs.

2.2.3 Fraud Detection Systems

A Fraud Detection System is a service offered by specialized companies to identify fraudulent CNP and/or CP transactions, such as in e-commerce. They make real-time decisions based on machine learning knowledge and on human intelligence and expertise.

These systems analyse extensive amounts of information, from several sources, to make a thoughtful recommendation. Commonly, among the analysed data is: IP geolocation, proxy detection, chargeback blacklists, social graphs, issuing-bank data, and address, phone, and email verification [10]. Mobile data can also be assessed by tracking the online fingerprint from customers and by determining their activity patterns [10].

This kind of analysis is thorough and quick. Some services can automatically accept and block transactions, according to a set of triggers and rules established in the system. Others just provide a recommendation or score, which helps the user to make a final decision over the transaction. Some automatic services claim to reduce manual review time by 80 percent [53].

Summing up, the use of a Fraud Detection Systems enables the e-Commerce company to reduce the costs associated with: extensive manual reviews, fraudulently ordered merchandise, friendly fraud, international order decline rates, chargeback fees, card association penalties and fraud related customer service calls [53].

2.3 Data Analysis Techniques for Fraud Detection

In this technological era, more and more business data is being managed and stored in IT systems. Every day, large amounts of data is gathered by companies to support their business processes. Often, fraud cases get hidden in those large volumes of data [6].

Data analytic techniques can play a fundamental role in the early detection and prevention of fraud [22]. By scanning through all gathered information, it is possible to derive some insight to detect potential instances of fraud, which enables the implementation of effective fraud prevention programmes [5].

However, significant progress in fraud detection can be made by looking at the data beyond the individual data points, in order to understand the relationships between them [30]. Often, this doesn't imply gathering new data nor more quantity; instead, it is necessary to use different tools, capable of underlying the hidden connections and patterns on data.

By applying data analysis techniques, useful information can be uncovered, to support decision-making regarding potentially fraudulent transactions. Link analysis is one of those techniques and it can help to expose previously undetected fraud, the entities committing it and the connections between those entities, when existing.

2.3.1 Link Analysis

Link analysis is a data-analysis technique used to explore associations between different objects. By comprehending the associations behind complex webs of evidence, it is possible to draw conclusions that are not apparent from the analysis of the raw data [40, 56]. Therefore, this technique is widely used in fraud investigations, either to detect fraud, uncover criminal networks or to expose fraudulent schemes [9, 85].

The results obtained by link analysis constitute a visual reference that, somehow, disclose the relationship between the data points. Typically, linked data is modelled as a graph in which the objects are represented as nodes and the links denote the relationship between them [47]. Both the nodes and the links can have associated attributes, specific to the domain.

Any type of data that can be sorted and tied together can be processed through link analysis. A lot of useful information can be learned about a network and its nodes by studying its links. However, looking at data relationships isn't exactly straightforward and configuring a link analysis can be a major endeavour [77, 54].

2.3.2 Components of Link Analysis

Two components can be distinguished in link analysis: (1) the generation of the links between the graph nodes and (2) utilization of the resulting linked graph [65].

Link Generation comprise the process of computing the nodes, their connections and respective attributes. Different approaches can be used to define the links, resulting in different types of linked graphs. Among them:

- Explicit Links: a link is created explicitly between related entities;
- Aggregate Links: several explicit links may be collapsed into a single link, creating an aggregate link;
- Inferred Relationship: links created between pairs of nodes according with inferred strength of the relationship between them, as computed by specific algorithms or rules.

Once a graph has been defined starring the nodes, links and attributes in analysis, it can be browsed, searched or used as part of a decision system, in order to draw conclusions.

2.3.3 Entity Linking

Entity Linking, otherwise known as Entity Resolution, is the process by which all records related to an entity are aggregated into a cluster [38, 73]. Depending on the objective of the analysis, the element designated as entity can vary. For example, an entity can be a customer, a merchant, a issuer, a credit card, an ATM, a phone number, an address, etc [23].

The identification of entities across data sources is vital in several fields. It can be used to find duplicate information in mailing lists and bibliographic databases, perform price comparisons, analyse publications via author referrals and even can be used to identify the same customer interacting with a company using different contact information [88, 41]. Entity linking is, therefore, used to overcome errors and inconsistencies in databases, determining which registers are likely to be related with the same entity [95].

Another area of interest for Entity Link analysis is criminal analytics. Fraudsters often provide false identification data. Among that deceitful information can be found addresses of acquaintances, personal information of deceased persons and even fake data. Therefore, in fraud detection domain, variations and errors do not occur due to faulty data entries or to the changing nature of people's personal details, but because fraudsters deliberately modify their personal information in order to conceal their true identification [24]. Link analysis techniques can be used to assess if those deceptive identity details do refer to a real person, or not, and to connect entities through those attributes.

This type of analysis goes beyond transactions and customer views. The goal of Entity Linking is to analyse activities and relationships within a network of related entities, in order to identify patterns of behaviour that come across as suspicious when analysed across related entities, and to uncover potentially associated entities, accessing the presence of a criminal conspiracy [14].

2.4 Data Visualization

Data visualization is the depiction of data encoded as a visual object, such as a pictorial object or a graphical format. Computer-based visualization systems provide visual representations of datasets. The main goal of a data visualization is to represent the data in a clear and efficient way, in order to provide an insightful interpretation of that data[18]. Difficult concepts become easier to understand in a data visualization and hidden patterns emerge.

The concept of visually represent data in order to understand it has been around for centuries. The human brain processes information more easily and quickly when

it is presented in a pictorial or graphical format, rather than presented or listed analytically [79]. Furthermore, in a data visualization, the information is displayed in a universal manner, easy to understand and to share with others [20].

There are several ways to display the same information, and each one highlights different angles of the same information. It is important to identify and understand the story behind the data that resonates both intellectually and emotionally with the target audience in order to select the proper visualization to best deliver the intended message [93]. A meaningful story is a powerful way to convey a message.

A good data visualization must be easy to understand but, simultaneously, it must be thought provocative. The use of layered data allows to convey a more complex message without becoming too distracting. Moreover, when taken out of context, the data visualization must still be understandable through the story it contains.

2.4.1 Data Types

At the data level, the type of the data represents its structural or mathematical interpretation. Five basic data types can be considered [55]:

- Item: discrete individual entity, such as a row table or a node in a network. In this project an item constitutes a financial transaction.
- Attribute: also known as variable or data dimension, is a specific property that can be measured, observed or logged. Transactions have several related attributes such as customer ID, merchant ID, timestamp, etc.
- Link: relationship between items, typically within a network.
- Grid: sampling strategy for continuous data that takes in consideration geometrical and topological relationships between cells.
- Position: spatial data, which provides a two-dimensional or three-dimensional location in space.

2.4.2 Attribute Types

At the highest level, two kinds of attributes can be distinguished: quantitative and qualitative attributes [62]. Quantitative attributes work with measurable variables while qualitative attributes deals with characteristics and descriptors that cannot be measured, but can be observed subjectively. Therefore, numeric variables contain numeric values that describe a measurable quantity, while categorical variables describe a quality or characteristic of a data unit.

Numeric attributes can be further distinguished between continuous and discrete attributes. Continuous data represent measurements whose values can take any value between a given set of real numbers and can be divided and reduced to more precise intervals (for example, height, weight, age). Discrete data represent items

that can be counted and whose values van be listed but not fractionated (such as number of children in a family, number of cars, etc.).

Categorical attributes can be described as ordinal or nominal. Values taken by ordinal attributes can be logically ordered or ranked, originating a hierarchical category (for example academic grades, clothing sizes, etc.). Nominal attributes cannot be organized in logical sequences (some examples are sex, eye colour, religion, brands).

The relation between different types of statistical data is represented in figure 4.

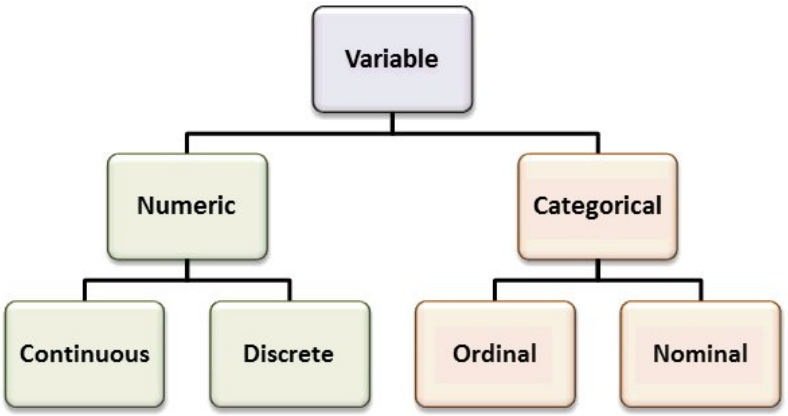

Figure 4: Flowchart representing different types of Statistical Data [62].

Some attributes have a hierarchical structure [55].

2.4.3 Dataset Types

A dataset constitutes a collection of data, which is the target of the analysis. It's made up from the five core data types. Four basic dataset types can be considered (figure 5) [55]:

- Table: a table is a familiar data representation made up of rows and columns. Each row represents an item of data and each column constitutes an attribute of the dataset. Therefore, each cell contains a value for the pair item and attribute it represents.
- Network: a network is used to represent relationships between items. Each item is represented as a node and a link denotes a relation between two items. Both the nodes and the links can have associated attributes.
- Field: a field dataset type contain attribute values associated with cells wherein each cell contains measurements or calculations from a continuous domain. This continuous data takes into account questions of sampling, interpolation and reconstruction of the data.
- Geometry: this dataset overlaps specific information about each item with explicit spatial coordinates.

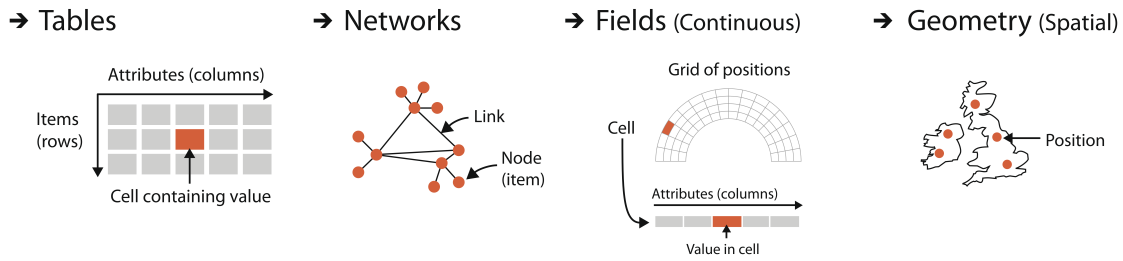

Figure 5: Basic dataset types [55].

Other ways to group information [55] are:

- Set: unordered group of items.
- List: group of items with a specific order.
- Cluster: group of items based on attribute similarity.
- Path: ordered set of links segments, connecting nodes over a network.

Often, complex combinations of these basic dataset types are found in the real world [55].

2.4.4 Stages of Data Analysis

One of the biggest challenges in data visualization is to find the best representation to display the multi-dimensional nature of the variables in analysis.

To obtain pertinent information from large amounts of data, the first step is to explore the dataset. On this exploratory phase, several different types of graphics are used in order to capture the structure of the data and to identify hidden patterns, trends and relationships between the items in the dataset. These graphics should be fast, informative and highly interactive.

Once the main aspects of the dataset are figured out, presentation graphics can be drawn, to be presented to a broader audience interested in the dataset. These graphics are conclusive and support the findings obtained in the previous exploratory phase. Often, they don't explain how a result was reached, only uphold the achieved conclusions without questioning. These graphics might not have much interactivity but are well thought and easy to read, in order to be understand by the masses.

While thousands of exploratory graphics can be built in the exploratory phase to support the work of a small group of data analysts, one presentation graphic is enough to disseminate their conclusions over thousands of people [55].

2.4.5 Exploratory Analysis

When analysing data, a good starting point is to search for patterns. Patterns in data can be found when analysing its distribution around the centre, how it is

spread and its shape (figure 6) [90]. Trends and correlations [51] are also interesting characteristics to be analysed, they can indicate a predictive relationship, which can be exploited in practice.

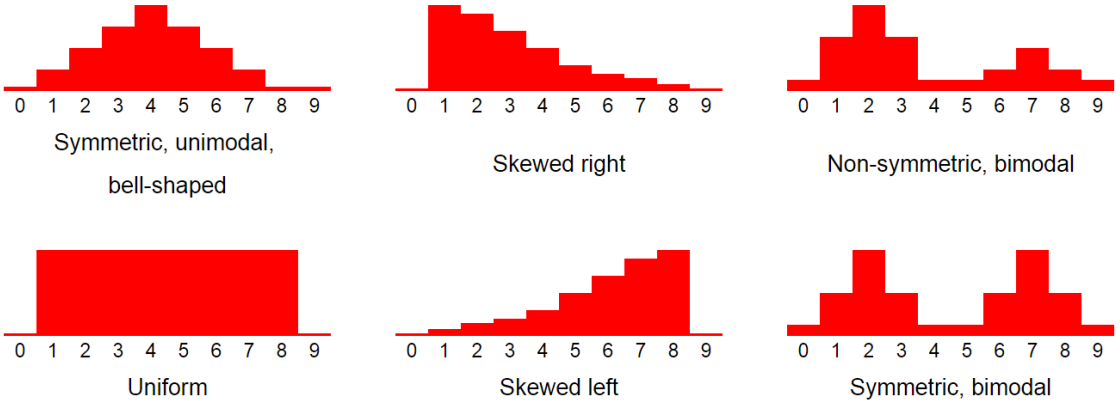

Figure 6: Representation of different distribution shapes [62].

Unusual features, such as outliers, can also provide insightful information about the dataset. An outlier is a point of data significantly different from the others. Frequently, their origin is caused by errors that need to be fixed. However, they may be the first indicator of a new data trend [39]. Figure 7 features a radar chart representing the number of visitors in a website across a day, grouped by the day of the week and stacked on top of each other. Daily trends are constant but for two outlier peaks around 4am and 11 pm every day of the week, where the number of visits far exceed the visits at any other time of the day. Further analysis concluded that this web traffic was being created by Web Crawlers and Spiders, which have scheduled their visits to this particular website at those time points [64].

2.4.6 Types of Data Visualizations

There are several types of data visualizations available [54]. Among those, are:

- Hierarchical Structures: ordered set in which the elements are organized according their relationships to one another. The nature of each relationship is variable, according to the field domain and type of the system. The best known example of this kind of visualization is the hierarchical tree.
- Relational Structures: this visualization focuses in relationships between items from the dataset. The most common visualization of this type is a network.
- Spatial Structures: spatial distribution of data according to the relative position of its components. The most common representation of a spatial structure is a map.

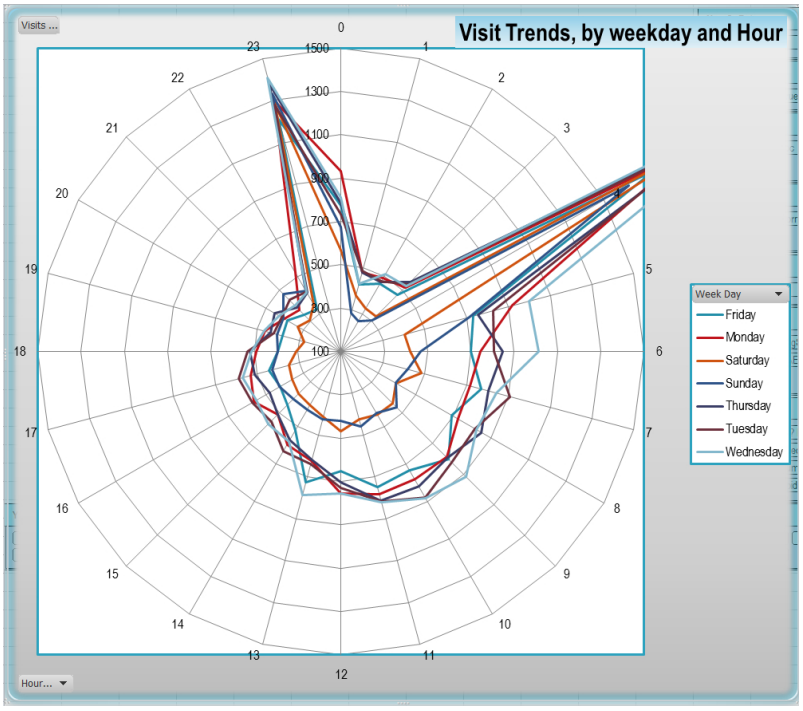

Figure 7: Radar chart representing the number of visitors at a given website. The number of website visits across the day is grouped by days of the week, which are stacked on top of each other [64].

2.4.7 Networks

A network is a simplified representation of a system that uses nodes to represent items and links to represent the relationship between two nodes. The structure of the network is designated as topology. Additional information can be added to the vertices and edges in a network, to capture more details about the system.

Networks can be represented using three different approaches: lists, matrices and node-link diagrams [54]. An adjacency list contains a complete list of links in a network, encapsulating the topology. However, lists can be quite unmanageable for large networks. The adjency matrix is a grid of nodes, in which cell represents the presence or the absence of a link between two nodes. Node-link representations are denoted as graphs.

The most common type of networks layout are: linear, Sankey type diagrams, force directed, circular, polar or radial, community structure, geography based and matrix [54]. An explanation of each layout is provided in figure 8.

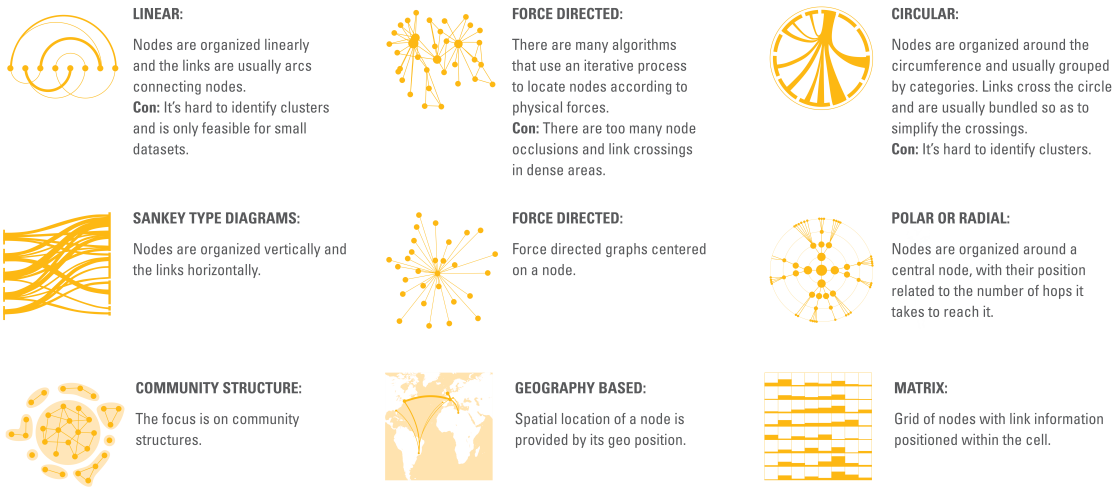

Figure 8: Most common types of network layouts [54].

3 State of The Art

In this section, several Fraud Detection Systems, with Alert Management functionalities for e-commerce, will be analysed and compared.

3.1 Alert Management Solutions

An Alert Management System is a type of event management system [16] used to organize and track alerts in a company or business [89, 37]. Different AMS can have different functionality such as: manage incidents [59, 78], intrusion detection [58], home automation [91] and fraud detection [32].

An alert is generated by Machine Learning or by a set of rules, defined by the system administrator, that specify the conditions in which it occurs. When those conditions happen, an alert is fired [94]. Default actions can also be designated to be performed when an alert is caught.

Alert queues can be defined to match business requirements. Then, when an alert is fired, it is placed in its respective queue, to further processing [94].

There are several Fraud Detection Systems, available for e-commerce, which offer Alert Management functionalities. In this section we will analyse the following systems: *Feedzai Cloud* [33], *Sift Science* [80], *Riskified* [74], *Signifyd* [84], *Merchant Protector* [72], *FraudLabs Pro* [71] and *Subono* [87].

3.1.1 Feedzai Cloud

Feedzai Cloud is a SaaS platform developed by *Feedzai* to provide merchants with a fraud detection solution. This service features a user interface to review alerts, the *Feedzai Cloud Dashboard*. Merchants can easily integrate with this solution, which is available on *Shopify* [82] and *Bigcommerce* [17].

This application automatically evaluates every business transaction that enters the system, scoring its risk of being fraudulent from 0 to 1000 — the higher the score, the higher the probability of an order being fraudulent. Using these values, transactions are classified in either *safe* or *suspicious*.

The application's *main page* (figure 9) presents two content panels: *Transactions Overview* and *Transactions List*. The *Overview* panel display the number of *Suspicious*, *Safe* and *All* transactions from the last 30 days, using a colour scheme to represent each transaction type, and the amount of money involved in each category. In addition to this information, a static bar graph represents the distribution of *Safe* and *Suspicious* transactions throughout the time. On the left, a list of the most recent transactions performed is presented. Clicking in any of them redirects to the *Payment Details* page for that transaction.

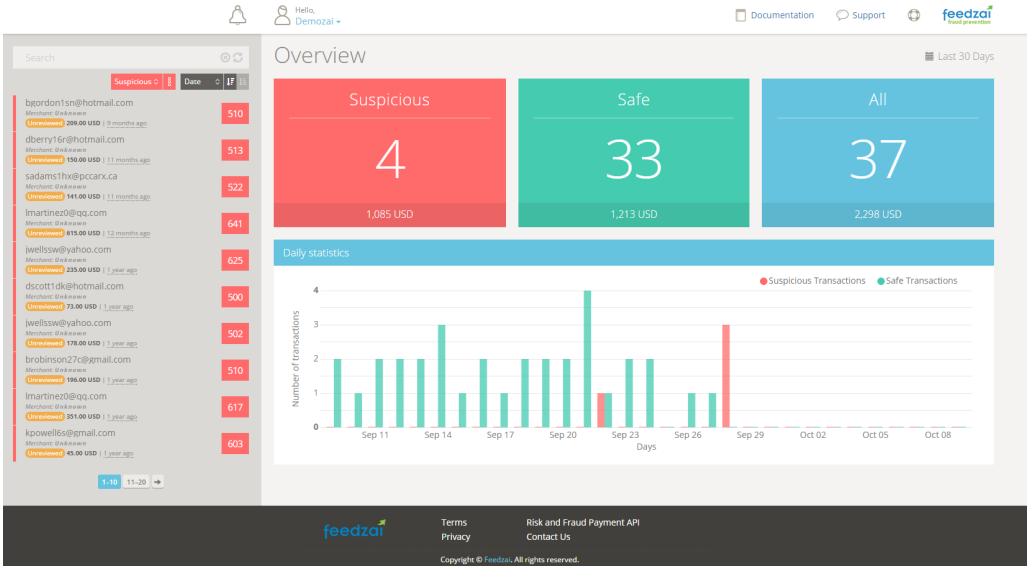

Figure 9: *Feedzai Cloud Dashboard* main page.

The *Payment Details* page (figure 10) shows the details of the selected order. This dashboard is divided in three major sections: *Transaction Details*, *Payment Details* and *Predictions*.

The section *Transaction Details*, on the header of the page, contains the main details from the transaction, such as: user e-mail, amount and score, given by the Machine Learning Fraud Models and rules. Additional information about the user can be checked by clicking in a button, which opens a pop-up detailing information such as: phone number, device ID, IP address and credit card used.

On the *Payment Details*, a selection box allows the user to mark each transaction as *Fraud* or *Not Fraud*. This empowers the user to teach the Machine Learning Fraud Model how to adapt to fraud patterns observed in his store.

On the centre of the *Payment Details* section, specific order information is displayed. At the right, information about the transaction, such as the time it occurred and its content, is shown. On the left, geographical data relative to the transaction is presented, namely: customer location (based on his IP address), customer address, shipping address and billing address. This information is also represented on a map. Additional data regarding the transaction, including the merchant and the client involved in it, is shown in a pop-up, by clicking on a button.

In the lower part of the *Payment Details* page, Machine Learning Fraud Model predictions are listed, grouped by *Interesting Facts*, *Reasons to Allow* and *Reasons to Block*. Each fact contains three components: description, a meaningful explanation describing its relevance for the selected transaction; risk factor, which represents the risk of accepting a transaction with this pattern in comparison with the bulk of all other transactions; confidence, which represents the frequency of occurrence of this pattern, expressing the Fraud Model's accuracy. These facts provide valuable insights to the user when the legitimacy of the transaction is hard to confirm.

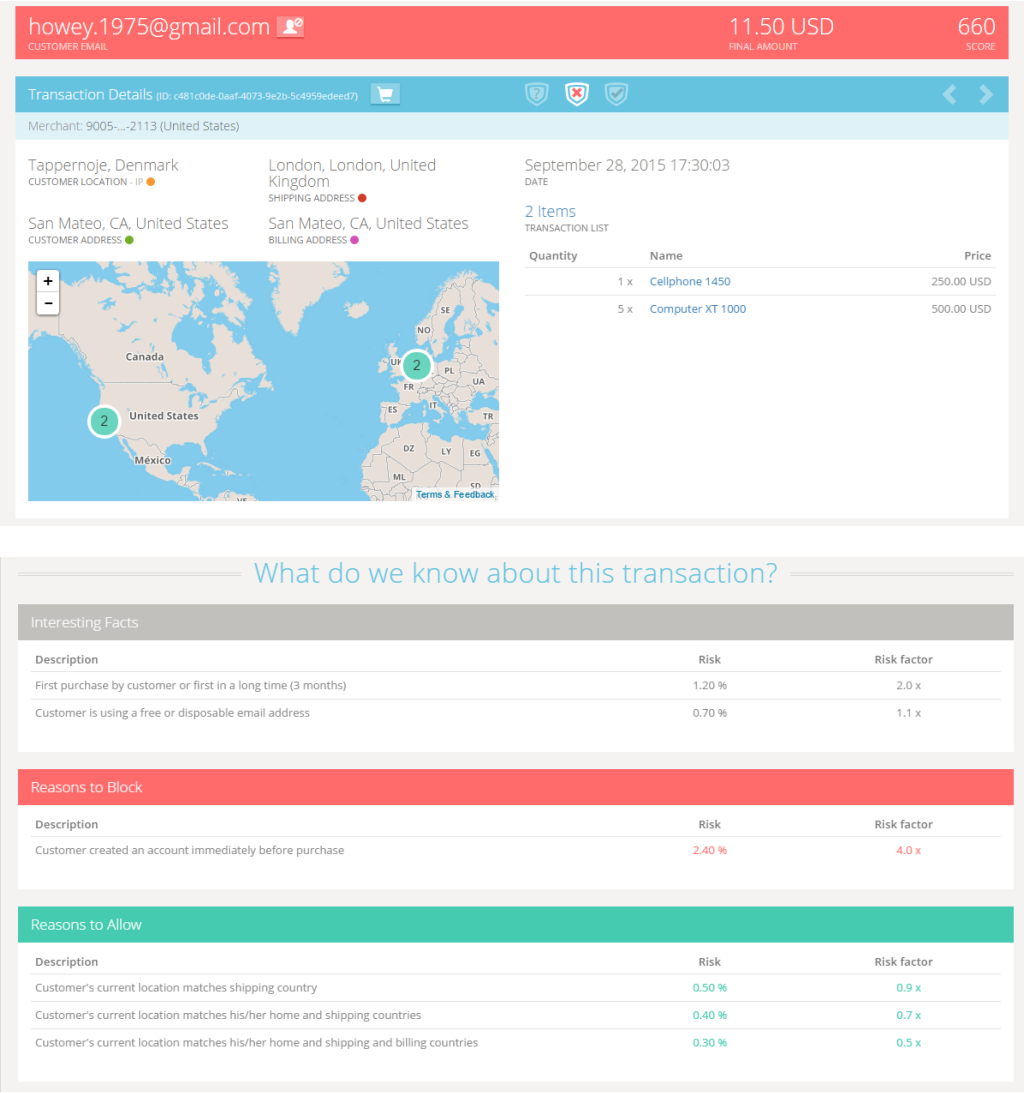

Figure 10: *Payment Details* of a given transaction.

3.1.2 Sift Science

Sift Science uses machine learning technology to detect fraud and their solution offers customization for each business and real-time learning. Their approach to machine learning leverages thousands of different signals to detect fraudulent behaviour and, when there is need for manual review of a transaction, generates an alert and notifies the user via e-mail or HTTP notifications. Furthermore, this SaaS platform partners with multiple third party data providers to keep track of social media profiles, disposable email domains, IP address geolocations and other interesting factoids.

The application's *main page* lists all transactions and, using a set of filters, allows users to create customized lists of users or orders. Selecting a transaction redirects to the *User Details* page of the customer who performed the transaction.

The *User Details* page (figure 11) provides investigative tools and critical information about a user. It is organized in several tabs: *Attributes*, *Orders*, *Identity*,

Locations, *Network*, *Signals* and *Activity*. Other information is also on display, such as the user score, date of last activity and status (bad or not bad). It also provides a mechanism to manually alter the user's status.

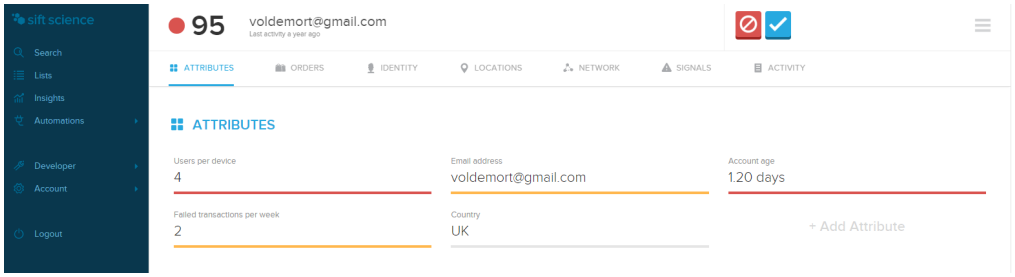

Figure 11: Sift Science's *User Details* page, featuring the *User Attributes* section.

The *Attributes* section displays *user information* and *attributes*. Among the *user information* there are personal details such as email address, country and account age, while the *attributes* contain information such as users per device and number of failed transactions per week. The *Orders* section list all the user's orders, including details such as: id, amount, date, payment type, status and number of transaction attempts. The *Identity* section consolidates the user's basic identity information, listing all the names, phone numbers and email accounts linked to the user as well as his social data (Facebook, Twitter, LinkedIn, Google Plus, Pinterest, Skype, Yelp, Foursquare, Yahoo, Gravatar). On the *Location* section, the shipping address, billing address and IP location of the user is presented and represented on a map. The *Signals* section lists the most suspicious fraud signals that *Sift Science* found for this user, presenting a risk factor for each signal that evidence it likeliness amongst fraudsters.

The most interesting sections from the *User Details* page are the *Network* section (figure 12) and the *Activity* section, (figure 13) since the information is displayed using diagrams to enrich the information on display.

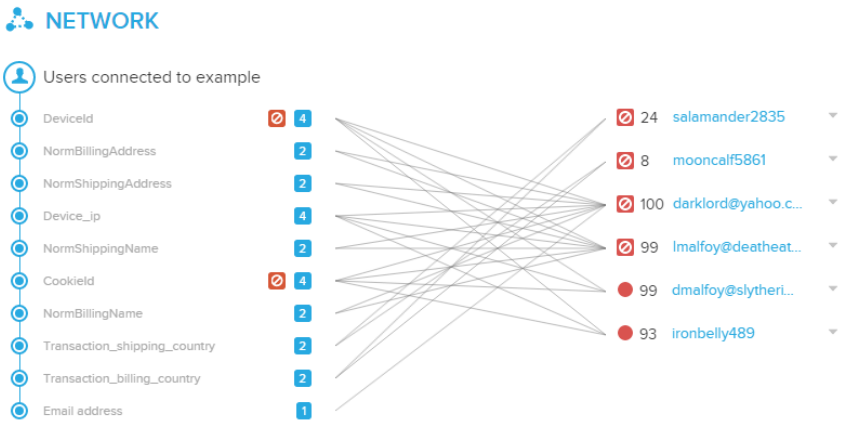

Figure 12: Network diagram from Sift Science's *User Details* page.

The *Network diagram* uses *Link Data Analysis* to display connections between shared IPs, devices and domains, in order to identify fraud rings. Clusters of users

(especially bad users) are indicative of that. However, this kind of visualization can become cumbersome.

The *Activity diagram* displays all actions and events that *Sift Science* has registered for the selected user, listed in chronological order, offering a timely perspective of the user's actions since he created an account in the system.

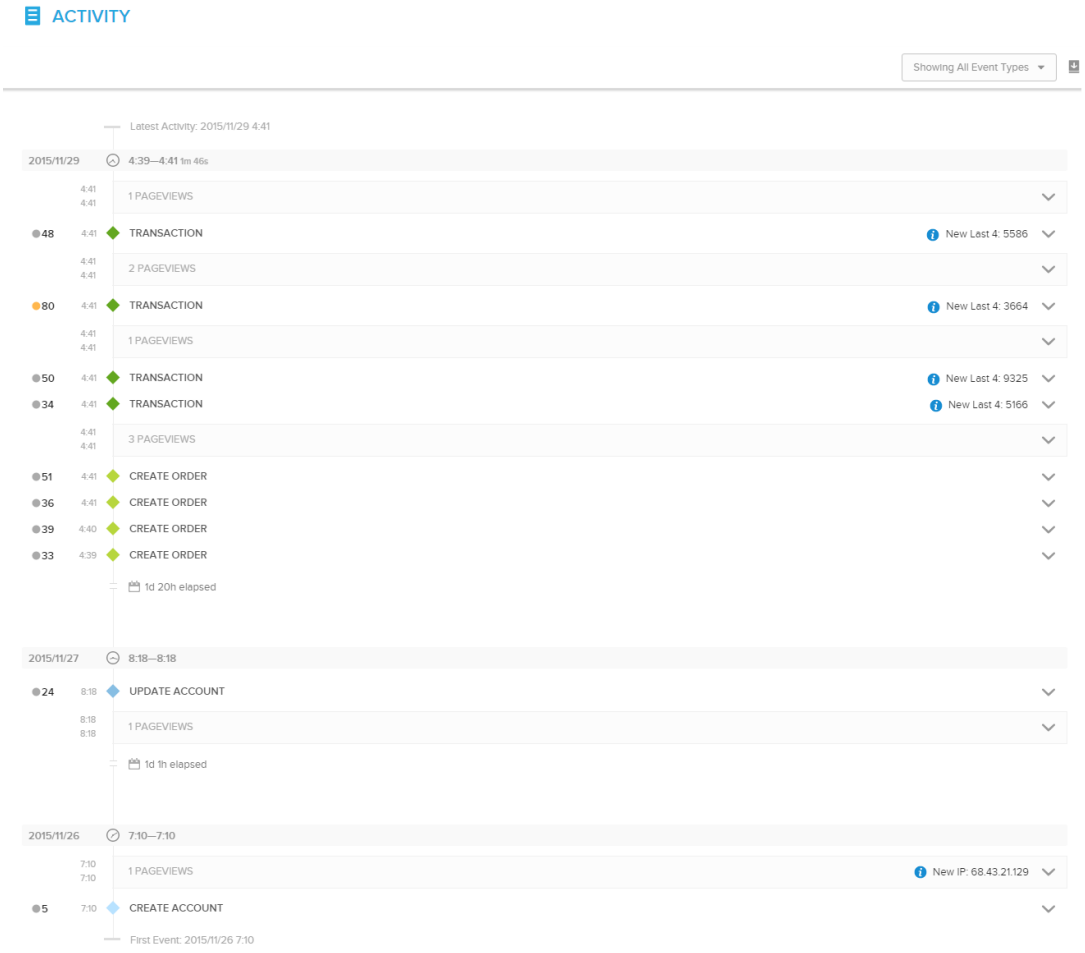

Figure 13: Activity diagram from Sift Science's *User Details* page.

3.1.3 Riskified

Riskified is an end-to-end fraud prevention solution for online merchants with a different philosophy: all the orders sent to the system are reviewed by the company and *Riskified* provide chargeback guarantee over the reviewed orders. The merchants can select which orders are to be reviewed: they can either choose to review all orders or only the international orders or only a set of hand-picked orders that they believe to be high-risked. Therefore, their UI is very simple.

Real-time data and human insights are continuously fed into *Riskified* machine learning models and every datapoint from *Riskified* ecosystem is linked, to deliver a self-

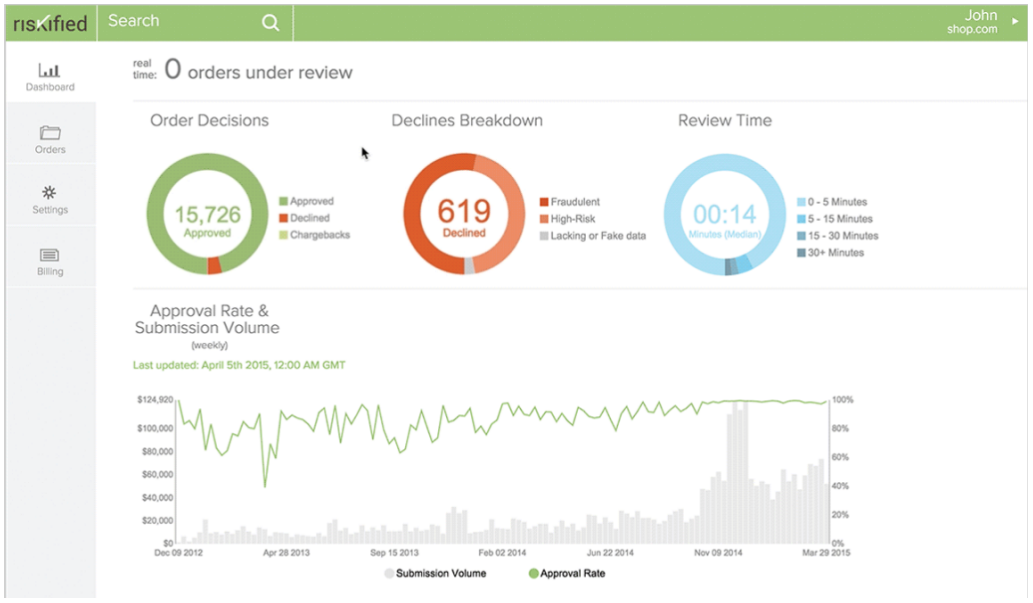

Figure 14: *Riskified Web Application* dashboard.

optimizing solution to detect fraud. Each submitted order is reviewed using multiple models and data inputs to obtain an actionable *approve* or *decline* decision.

The *Riskified Web Application* (figure 14) displays a dashboard to help the user to understand *Riskified*'s decisions. Data visualizations, in the form of circular ring graphs, illustrate approval and decline rates and the reasons why the orders are being declined. In the bottom, a line graph represents order trends, considering the volume of orders submission and its acceptance rate.

The *Order Data Tool* (figure 15) allows to the user to automatically or manually submit orders for review. Real-time order status can be followed and risk indicators, as well as the reasons why the order was approved or declined, are displayed.

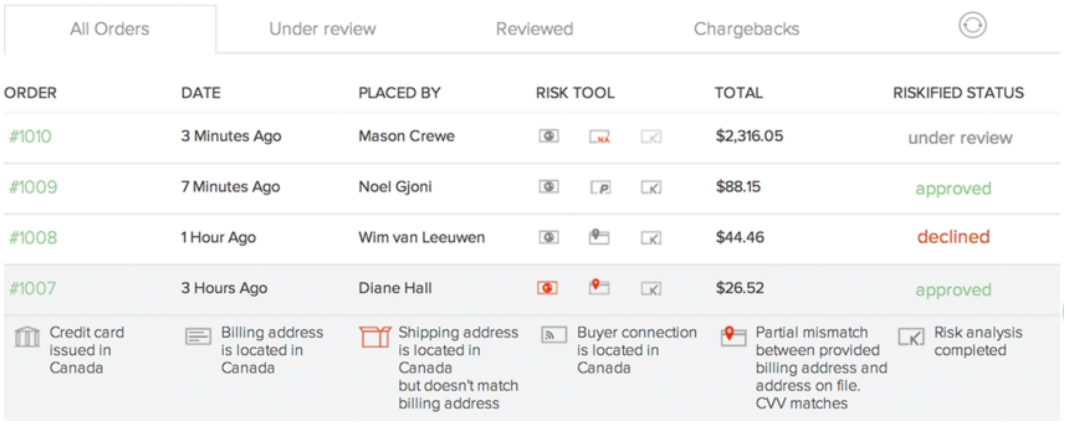

Figure 15: *Riskified Order Data Tool* featuring an accepted order and displaying the reasons of its approval.

3.1.4 Signifyd

To combat fraud, *Signifyd* resorts to robust and scalable enterprise systems and processes that combine several datasources to produce signals, which determine if a transaction is good or bad. Among those sources are: phone number, address, device fingerprints, IP geolocation, proxies, social graph, network of thousands of sellers, issuing banks, cross-merchant blacklists, velocity, search engines and public records. Other features from this system are real-time scoring, multi-channel coverage and shared network. All of *Signifyd* customers benefit from a shared blacklist.

Users can choose between *on-demand* or *complete order* review. Every decision comes with a report that includes a score and a detailed explanation about the advice, which can be *decline*, *review* or *accept* the transaction.

Signifyd console (figure 16) is comprised of a *Case Queue* panel and *Case Report*. The *Case Queue* panel lists cases from the system. The *Case Report* displays information about the currently selected case. On top, it is displayed a code coloured analysis of the user's address, device and email. The *Summary* section lists payment, account and order information and displays a map with geolocation data.

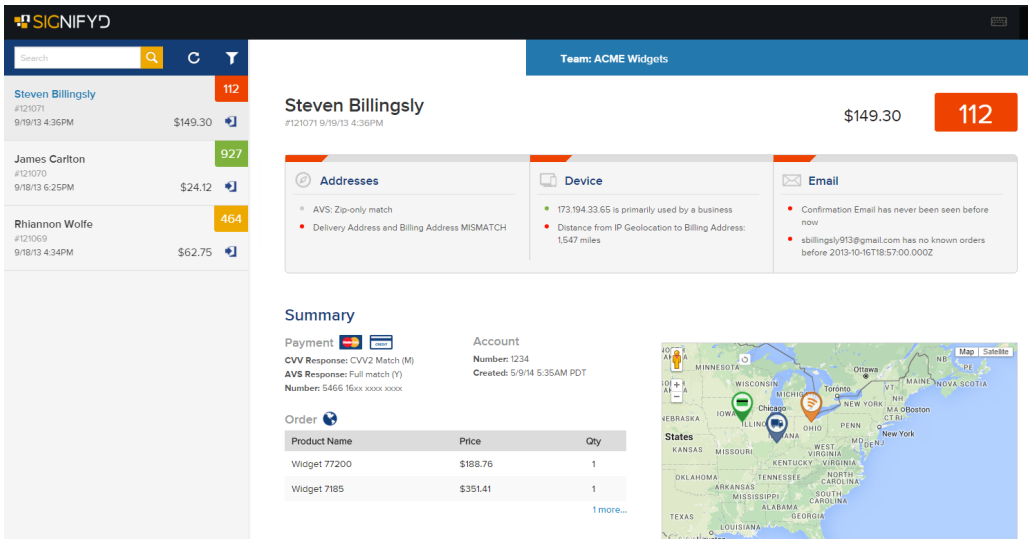

Figure 16: *Signifyd* console.

Below this section, a more detailed analysis about how *Signifyd* score this order can be found (figure 17). Information about the user, its location, social profile, etc. is displayed.

In order to match information and connections, when an item is scrolled over, related information is highlighted. This is shown in figure 17.

Blue coloured links (figure 18) can be clicked to view the source from which the information was derived.

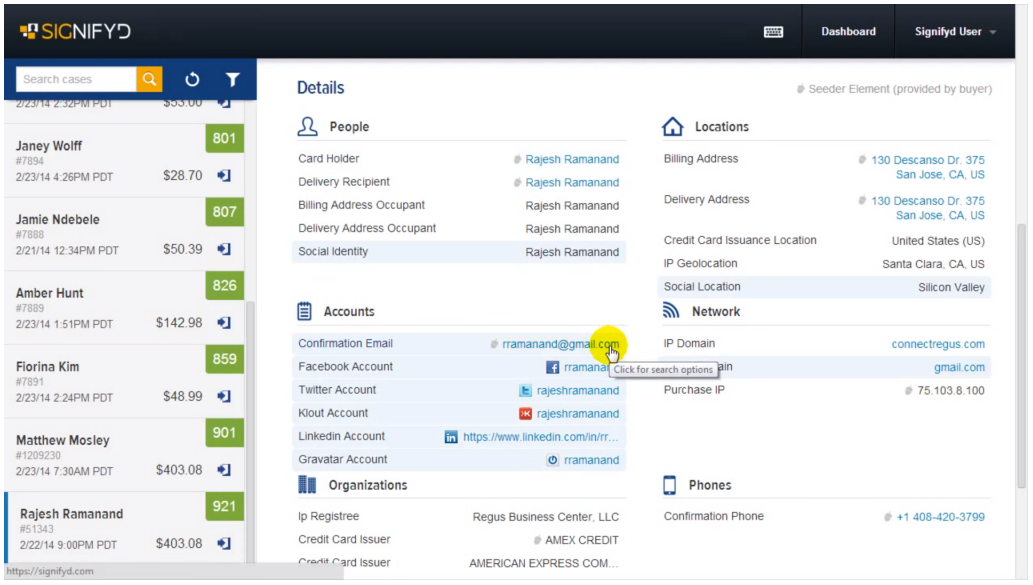

Figure 17: *User details* on *Signifyd* console highlighting information related with the field being scrolled over.

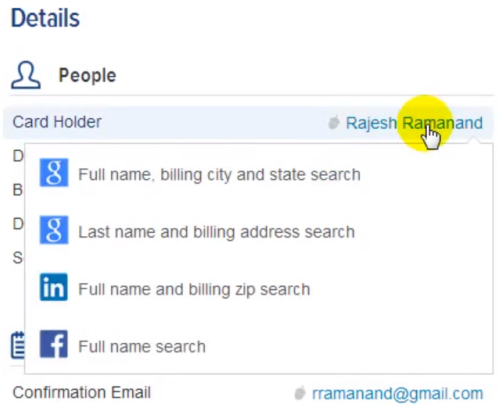

Figure 18: Dropdown menu displaying source information about the clicked field on the *Users Details* section on *Signifyd* console.

3.1.5 Merchant Protector

Merchant Protector is a SaaS platform that offers customizable order confirmations. Every order is automatically analysed and, with the click of a button, the user can request additional information from any customer. Two versions of the application are available, with more features being available in the complete version.

The *User Details* available on this application are: social customer data, fraud alerts API with explanations, history and purchase patterns and location mapping. Orders that share the same email, IP address, billing or shipping address, or customer name are displayed in the history and purchase patterns, as similar fraudulent orders are displayed when encountered.

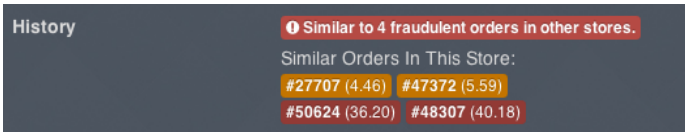

Figure 19: *History* Section from *Merchant Protector* application featuring similar fraudulent orders to the selected order.

3.1.6 FraudLabs Pro

FraudLabs Pro analytic engine analyses transactions parameters and reports fraud analysis in less than a second. Merchants can immediately decide their next action, based on a fraud distribution score or in custom rules by conditions. An interactive form is provided to allow users to submit their own custom rules, deciding what to do on each case (figure 20).

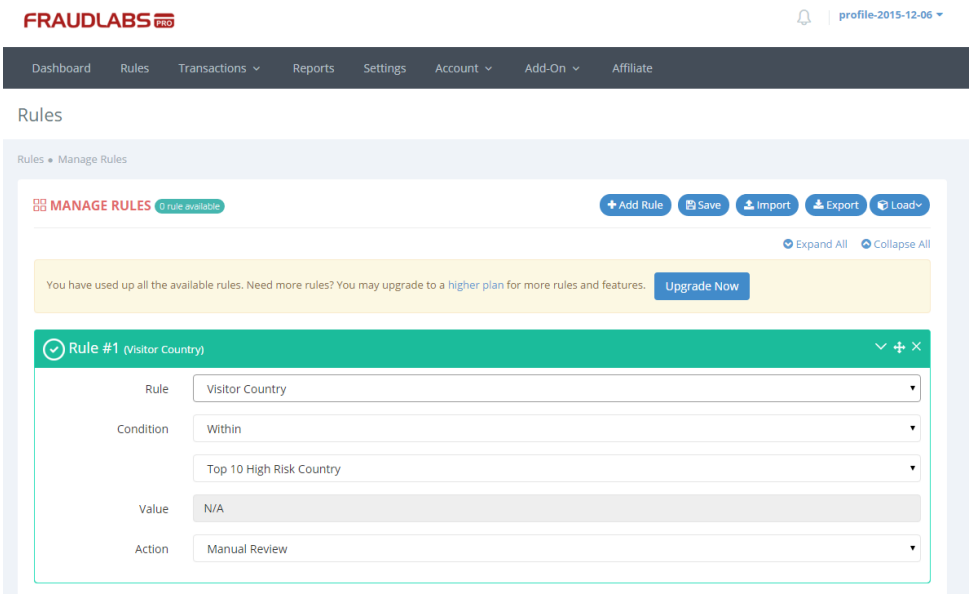

Figure 20: *Manage Rules* form in *FraudLabs Pro* solution.

FraudLabs Pro runs a thorough validation of each transaction to effectively detect a malicious fraud. It also validates the user, screening his IP address, email address, credit cards and devices, based on moderated crowd-sourcing data.

On the *dashboard* (figure 21), total number of *Approved*, *Rejected* and *Pending Review* orders is displayed, using a code colour scheme. An overview of the number and amount of the three order's types from the last 30 days is displayed. The number and amount of orders, according with their type, from the last 6 months is presented in a bar chart.

The *Report* tab presents a bar chart and a linear chart of the number of orders and amounts from the different order types in a period of time determined by the user.

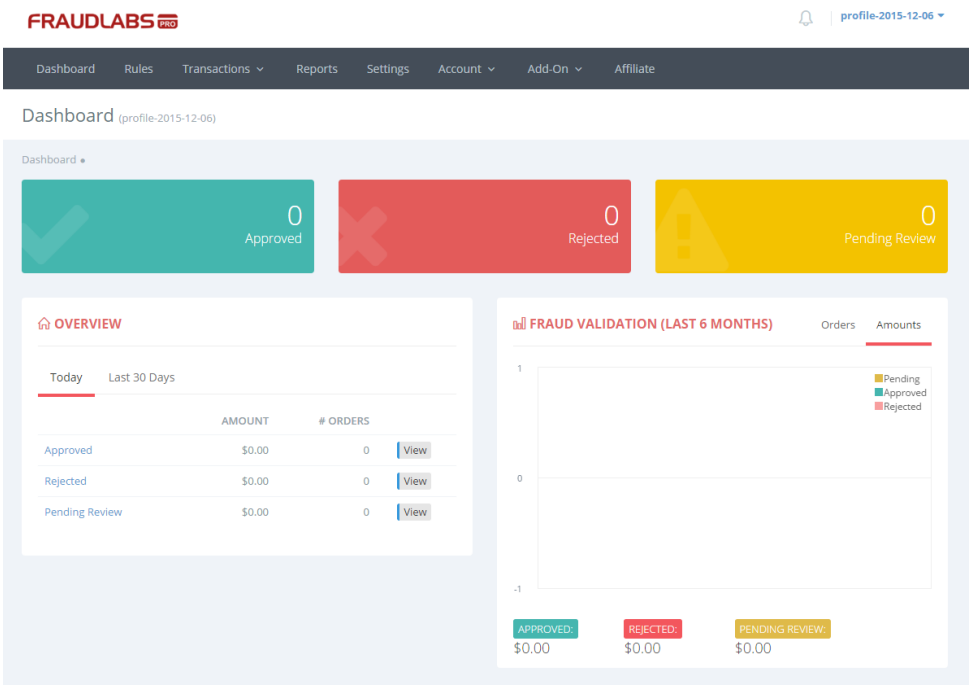

Figure 21: *FraudLabs Pro* main dashboard.

3.1.7 Subuno

Subuno flags fraudulent orders automatically with risk scoring that analyses over 100 risk factors. It works in a pay-per-use philosophy: the user sends the information to be reviewed, and each order is checked against risk factors and business rules such as a fraud score, block/white lists, velocity and order characteristics. The rules can be viewed, created, and updated by the user through *Subuno*'s application.

This system combine data from different tools to create a unique set of fraud prevention rules. Each order's individual detailed page offers colour warnings and a wide selection of tools to perform further verification (figure 22). *Subuno* consolidates over 20 fraud screening tools on to the same page to help the user to quickly assess and confirm flagged fraudulent transactions.

Among the information analysed by this application is: location where the customer placed the order from; estimation of how old the email address is; matching of the customer's name, address and phone number; customer used different shipping address; shipping request was placed overnight; shipping request is a high order value; verify if the customer is on the user's block list; customer social network information (Foursquare, Flickr, Gravatar, Angellist, Vimeo, Facebook, LinkedIn, Spokeo, Google, Pipl, Paypal).

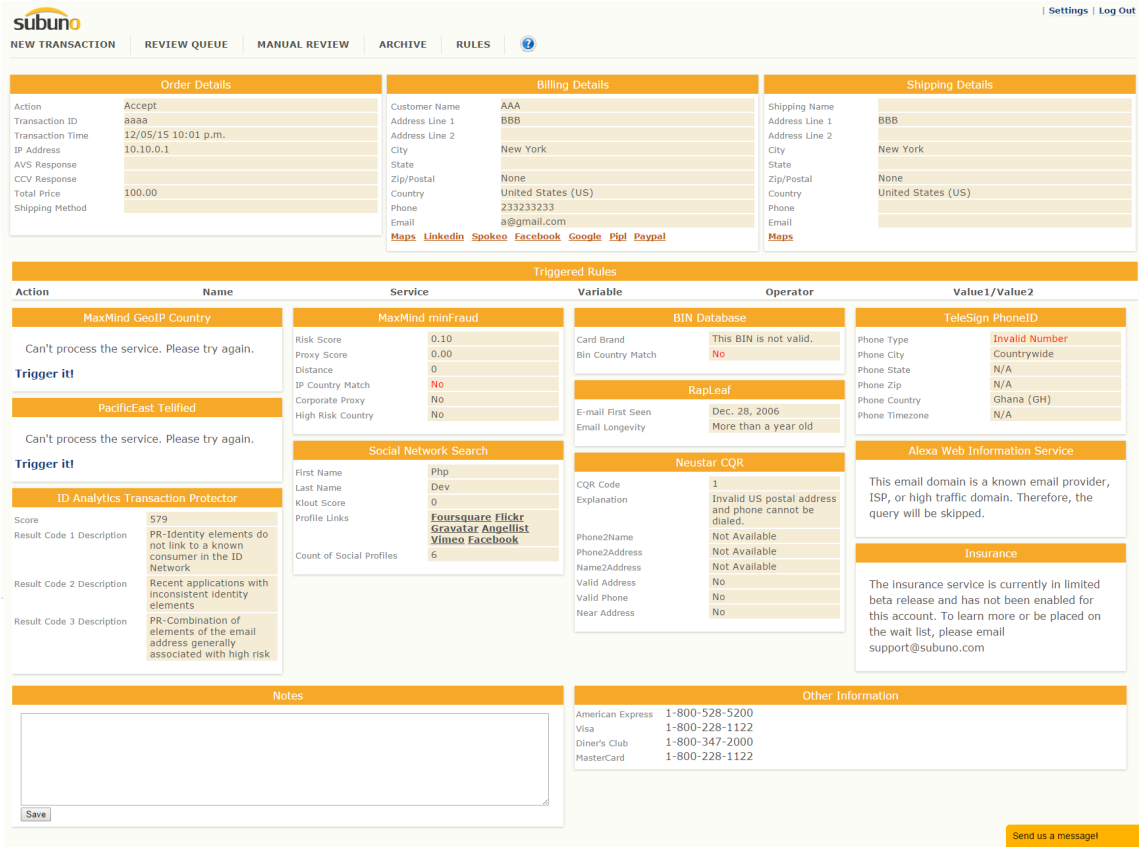

Figure 22: *Subuno* transaction analysis.

3.2 Comparative Analysis

This section aims to compare the Fraud Detection Systems with Alert Management capabilities presented on section 3.1. For that, it was necessary to define a set of metrics. The systems were analysed accordingly and the results from this comparison are presented and further discussed.

3.2.1 Metrics

In order to compare the solutions reviewed previously, it was necessary to define a set of metrics. Those metrics were grouped in four different categories, which are: Overview, Operation, Transaction, Customer. Each group definition and respective metrics are listed below:

- **Overview:** these metrics are related with the overall functioning of the system;
  - **Number:** The system displays the number of total transaction analysed and the number of transactions marked as fraudulent;
  - **Statistics:** A visual representation is used to present statistical information about the total number of transaction and the number of transactions

marked as fraudulent over time;

  - **Colour Scheme:** The system uses colour scheme to differentiate fraudulent from non-fraudulent transactions;

- **Operation:** this group of metrics represent how the system works and is organized;
  - **List Users:** The system has a list of analysed users;
  - **List Orders:** The system has a list of analysed transactions;
  - **Manual Decisions:** User can manually decide over a transaction;
  - **Emit Alerts:** When a transaction is flagged as fraud, an automatic alert is emitted;
- **Transaction:** group of metrics related with a transaction;
  - **Order Info:** Order information such as the items ordered, their quantity and amount is displayed;
  - **Payment Info:** Customer's payment information is presented;
  - **Score:** Score automatically attributed to the transaction is shown;
  - **Signals:** The system list the signals raised by Machine Learning Fraud Model predictions and their associated risk;
- **Customer:** group of metrics related with the customer who performed the transaction;
  - **Customer Info:** Customer information is displayed;
  - **Identity:** Customer's authenticity is somewhat verified (checking if email provider is reliable, email account age, phones number validity, etc.);
  - **Social Data:** Social data information about the customer is analysed;
  - **Geo-location:** Geographical information about the transaction is displayed and how;
  - **Network:** The system uses Link Analysis to investigate the network around the customer and detect fraud rings;
  - **Orders History:** History of the orders placed by the customer is listed;
  - **Activity:** Customer's activity on the system is presented;

3.2.2 Comparison

Comparison of the different Fraud Detection Systems with Alert Management capabilities, according with the metrics previously defined in section 3.2.1, is presented on table 1.

		Feedzai Cloud	Sift Science	Riskified	Signifyd	Merchant Protector	FraudLabs Pro	Subuno
Overview	**Number**	✓	✓	✓	✗	?	✓	✗
	Statistics	✓	✓	✓	✗	?	✓	✗
	Colour Scheme	✓	✓	✗	✗	?	✓	✗
Operation	**List Users**	✗	✓	✗	✗	?	✗	✗
	List Orders	✓	✓	✓	✓	?	✓	✓
	Manual Decisions	✓	✓	✗	✓	✓	✗	✗
	Emit Alerts	✓	✓	✗	✓	✓	✓	✗
Transaction	**Order Info**	✓	✗	✗	✗	?	✗	✗
	Payment Info	✓	✓	✗	✓	✓	✗	✗
	Score	✓	✓	✗	✓	✓	✓	✓
	Signals	✓	✓	✓	✓	✓	✗	✓
Customer	**Customer Info**	✓	✓	✗	✓	✓	✗	✓
	Identity	✗	✓	✗	✓	✗	✓	✓
	Social Data	✗	✓	✗	✓	✓	✗	✓
	Geo-location	✓	✓	✓	✓	✓	✓	✓
	Network	✗	✓	✗	✗	✗	✗	✗
	Orders History	✗	✓	✗	✗	✓	✗	✗
	Activity	✗	✓	✗	✗	✗	✗	✗

Table 1: Comparison between different Fraud Detection Systems with Alert Management Capabilities.
Fields marked with ✓represent a system which verify the correspondent metric, while fields marked with ✗represent a system which don't verify the correspondent metric. When no information was available, the field is marked with **?**

3.2.3 Discussion

From this comparative analysis, it becomes evident the lack of refined data visualizations in the alert management space. Current offers are more focused in the quantity of information to display and less in the quality, recurring mostly to textual information or simple graphical elements than to rich visualizations.

The systems on analysis present all a similar structure: the majority of them contains a sort of dashboard, with an overview of the fraud detection process, and a list of analysed orders, whose details can be further exposed. Graph bars, linear graphs and even circular ring bars are used to present the overview information of the fraud detection process. The remaining information is mostly listed in more or less organized sections or tables. Geo-location information is often represented in a map. Colour schemes are often used to distinguish between fraudulent and non-fraudulent transactions.

The most complete solutions are presented by *Feedzai*, *Sift Science*, *Riskified* and *FraudLabs Pro*. Systems like *Signifyd*, *Merchant Protector* and *Subuno* are more transaction-oriented and lack information about the overall fraud detection process, which is found in the other systems.

The majority of the analysed systems offer the user the possibility to make the final decision, regarding a transaction. In those cases, the systems display more and more complete information in their dashboards, in order to help the analysts to make an informed decision. Systems that lack this possibility, like *Riskified*, *Fraud Labs Pro* and *Subuno* tend to be more simplistic.

Some solutions use data enrichers to validate customer's information. Either by validating their personal information such as email address and phone number or by analysing their social data information, these systems use third part services to gather more information about the customer in order to confirm his identity. Among these systems are: *Sift Science*, *Signifyd*, *Merchant Protector*, *FraudLabs Pro* and *Subuno*.

Sift Science solution stands out from the remaining systems by the completeness of the available data and the adopted data visualizations. The main features of this system are: customer data is divided in logical sections; list the customer's historic of orders; customer's personal information is validated; displays customer social data. Furthermore, uses entity linking to investigate fraud rings and displays the results in a network graph. Moreover, all the customer's activity is presented on a temporal diagram.

4 Work Plan

This chapter introduces the development methodologies used on this project and explains how the planning evolved during each semester.

4.1 Methodology

Work on this project was quite different in the two semesters, therefore the development methodology used in each semester diverged. First semester was more oriented to investigation and followed the Waterfall development model. Second semester was oriented to the development and analysis of different data visualizations, and followed the agile methodology Scrum.

4.1.1 Waterfall Model

In the first semester of work, we opt to use, as the software development methodology, the waterfall model. Since this first stage was quite focused and static, the Waterfall development model was considered as the most suitable methodology for the first semester.

The Waterfall model methodology consists in a sequential design process, progressing through a series of phases in a linear fashion over time.

4.1.2 Scrum

Work on the second semester followed the *Scrum* methodology [45, 28], which is an iterative and incremental development methodology used in *Agile Software Development*. Adopting an *agile* methodology provides a flexible environment for software development, offering the ability to change requirement's priority and to adapt new features easily.

The *Scrum* framework structures development in cycles of work, designated as *Sprints*. *Sprints* have a fixed duration, usually from 2 to 4 weeks, and take place one after the other, without pause. At the start of an iteration, the team reviews the project requirements and selects the priority items to complete during the *sprint*. At the end of the iteration, the results are reviewed and a new *sprint* begins.

In *Scrum* there are three primary roles and the project responsibilities are divided among them. The *Scrum Team* is composed by:

- *Product Owner*: responsible to maximize the value of the product and the work of the Development Team;
- *Development Team*: group of professionals responsible for building the product under the supervision of the Product Owner;

- *Scrum Master*: responsible for the Scrum process, he helps the group to learn and apply Scrum in order to achieve business value.

The *Scrum Team* for this project is composed by Rafael Marmelo and António Alegria as *Product Owners*, João Miranda as *Scrum Master* and Inês Coelho as team member of the *Development Team*.

Work on this project was integrated in the Alert Manager Development team. Each *Sprint* had the duration of two weeks, with a Sprint Review, Sprint Retrospective and Sprint Planning at the end of each sprint. Besides that, daily meetings were held for the duration of this project.

In terms of software, *Confluence*[11] was used as the document management system and *JIRA Software*[12] was used to manage the project according with the *Scrum* framework.

4.2 Planning

4.2.1 First Semester

Work on the first semester focused on the elaboration of the dissertation proposal for this internship. This required a strong research component, in order to grasp the main concepts behind Alert Management for Fraud Detection, Entity Linking and Data Visualization. It also focused in the study of the market and competitors, as well as defining and understanding the requirements of this project and how to implement them.

At the beginning of the first semester, a high level plan was elaborated in which work was divided in several tasks, to be performed sequentially. These tasks are:

- **Initial Setup:** settle in Feedzai; get to know and experience tools and methodologies used in the company; get familiar with internal tools.
- **Background Knowledge in Fraud:** bibliographic review of the state of the art of fraud in e-commerce.
- **Feedzai Cloud Dashboard and Alert Manager:** analysis of uses cases and dashboards used in Feedzai Cloud Manager and in the Alert Manager application under development.
- **Data Visualization:** bibliographic review of the state of the art in data visualizations and the technologies used.
- **Competition Analysis:** analysis of available tools in the alert management for fraud detection domain.
- **Requirements Elicitation:** definition of user case stories, mapping of those stories into architectural drivers and architecture design.
- **Intermediate Report:** writing of the intermediate report.

- **Intermediate Presentation:** preparation and presentation of the intermediate presentation.

Distribution of those tasks over time is presented in the following Gantt diagram (figure 23):

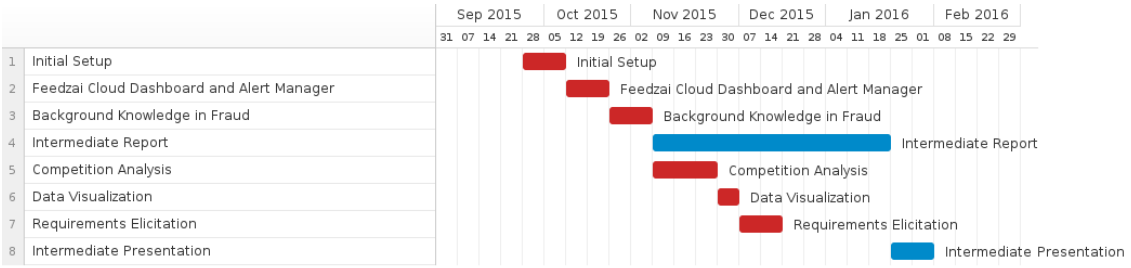

Figure 23: Gantt Diagram representing first semester work plan.

During this period, each week corresponded to 16 hours of work, which coincide with 16 ECTS, the internship workload for the first semester.

4.2.2 Second Semester

Work on the second semester was focused on entity link analysis and the development of data visualizations. The high level plan for this semester was:

- **Setup:** project restructuring; dataset exploration and testing; acquaintance with the development tools and project setup.
- **Data Visualizations:** design and development of several data visualizations for Entity Linking analysis with a demo database; exploration of different parameters, features and functionalities for each visualization. Featured visualizations were:
  - Table
  - Force Directed Graph
  - Circular Diagram
  - Chord Diagram
  - Geographical Referencing
  - Matrix
- **Exploratory tool:** building of an exploratory tool for Entity Linking, using real datasets and featuring the data visualizations previously built.
  - User Interface: design and development of the user interface
  - Data Visualizations Integration: integration of previously built data visualizations with the exploratory app

  - Database Refacturing: integration of the web app with datasets containing real data
  - Refinement: minor fixes and alterations to the application
- **Validation:** validation of the data visualizations, within the Entity Linking exploratory tool, through usability testing.
- **Consolidation:** consolidate the work done and analyse future directions.
- **Final Report:** writing of the final report.
- **Presentation:** preparation of the final presentation.

The following Gantt diagram represents the distribution of those tasks over time (figure 24):

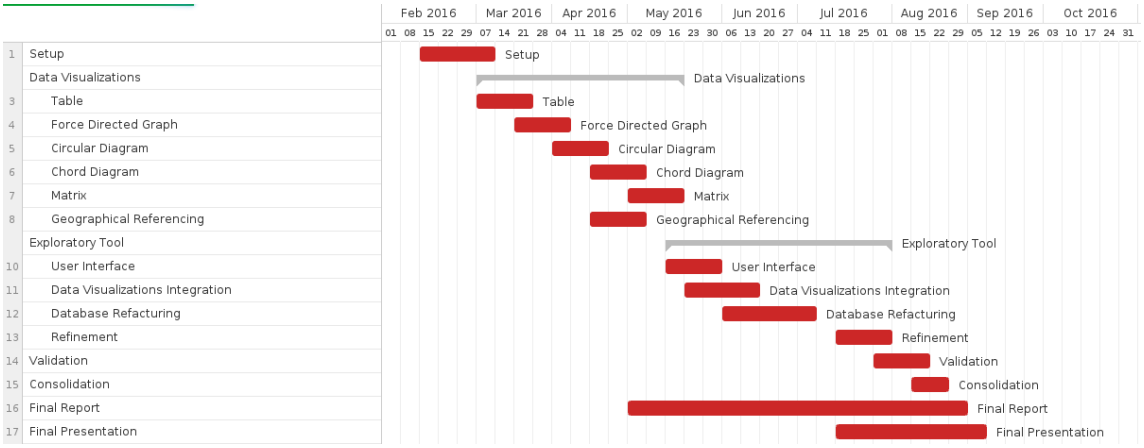

Figure 24: Gantt Diagram representing second semester work plan.

5 Requirements Specification

This chapter describes the Exploratory Tool web application to be developed, in general terms, and specify its requirements. The Overall Description subsection describe general factors that affect the product and its requirements. The Specific Requirements subsection present functional requirements, non-functional requirements and design constraints.

Information contained on this chapter was approved by the stakeholders of this project and was used as a reference to develop the system.

5.1 Overall Description

This section gives an overview of the whole system, providing a background for the requirement specification. It explains the process by which the preliminary requirements for this project were derived from the Alert Manager Application and defines the product and its functionalities.

5.1.1 Product Perspective

The Alert Manager Application, currently under development by Feedzai, aims to provide a User Interface to help fraud analysts to identify correctly, quickly and efficiently fraud and non-fraud cases, in conditions where an automated decision by a Fraud Detection System cannot be achieved.

To define the scope of this project, first we identified a series of user stories that reflect the general functionalities of the system. A complete description of this process can be found in Appendix A. The identified user stories were grouped in five modules:

I Overview;

II Entity Profile;

III Entity Linking;

IV Geographical Information;

V Geographical Profiling.

During this process, Entity Link analysis was disclosed as the most rewarding case to be developed for the Alert Manager Application. Therefore, work on this project focused on this subject.

The primary goal of this project was to create an exploratory tool for Entity Link analysis in the form of a web application. This application is able to read commercial transaction data from real datasources and build different data visualizations for Entity Link analysis. Those visualizations are highly interactive, permitting the

manipulation of several parameters, in order to fully explore the potentiality of each visualization for the dataset in analysis.

This web application was used as a testing tool for the validation process, in which several expert data analysts tested the different data visualizations, in order to select the most suitable one to figure in the Alert Manager Application and proposed alterations to improve its use. Of this process, resulted the proposal of an Entity Linking graphic to be integrated in the Alert Manager Application, after further handling.

5.1.2 Product Functions

The software system to be developed will be an Exploratory Tool Web Application for Entity Link Analysis for Fraud Detection. The web application will provide a series of highly interactive Data Visualizations to represent relationships between Entities performing commercial transactions, in order to visually identify fraudulent situations.

The system will be able to analyse data from different datasources and analyse relationships between a selected data entry and the remaining data in the dataset, according with first-order and second-order parameters, inputted by the user.

Besides those selection parameters, the user must be able to interact with the application to manipulate specific visualization parameters, in each available visualization, to improve the fraud detection experience for each case under analysis.

This system will be used as an exploratory tool for those data visualizations, with real data from e-commerce transactions, in order to select the best data visualization and the best conditions to use it.

5.1.3 User Characteristics

The target users of this system are fraud analysts, specialized in dealing with data analysis and fraud detection.

5.2 Specific Requirements

This section specifies the detailed requirements which the system shall meet.

5.2.1 Functional Requirements

The exploratory web application for Entity Linking in fraud detection shall fulfil the following functional requirements:

ID: FR1
Title: Select a Datasource.
Description: The application must be able to offer different datasource options for the user to select.
Dependencies: None
Priority: Desirable

ID: FR2
Title: Select a Data Visualization.
Description: The user must be able to select a type of data visualization to present the data, which shall be chosen from the following list:

- Table;
- Matrix;
- Geographical Referencing;
- Force Directed Graph;
- Force Directed Graph with Fisheye;
- Circular Graph;
- Chord Diagram.

When a selection is made, the visualization must be updated accordingly.
Dependencies: None
Priority: Essential

ID: FR3
Title: Select a Transaction.
Description: The user shall be able to select a transaction to be analysed, from the dataset.
Dependencies: FR1
Priority: Essential

ID: FR4
Title: Select the maximum number of transactions to display.
Description: The user must be able to define the maximum number of transactions to be analysed, at any given moment.
Dependencies: FR1
Priority: Desirable

ID: FR5
Title: Select a set of first-order parameters.
Description: The user must be able to choose, from a list of available parameters related with the datasource, which ones must be used as first-order parameters, to select a set of transactions related with the one under analysis.
Dependencies: FR1
Priority: Essential

ID: FR6
Title: Select a set of second-order parameters.
Description: The user must be able to choose, from a list of available parameters related with the datasource, which ones must be used as second-order parameters, to analyse relationships.
Dependencies: FR1, FR5
Priority: Essential

ID: FR7
Title: Get transactions from the database.
Description: Get transactions from the database, taking in consideration selected parameters by the user.
Dependencies: FR1, FR3, FR4, FR5, FR6
Priority: Essential

ID: FR8
Title: Build data visualization.
Description: With the data from the database, build the correspondent data visualization.
Dependencies: FR2, FR6, FR7
Priority: Desirable

ID: FR9
Title: Visually identify selected transaction
Description: The selected transaction must be distinguishable from the others transactions in the visualization, in order to be easily identifiable.
Dependencies: FR2, FR3
Priority: Essential

ID: FR10
Title: Alter selection parameters and update Visualization.
Description: Alterations in the selection parameters (transaction under analysis, maximum number of transactions, first-order selection parameters and second-order selection parameters) require the rebuild of the data visualization. with new data
Dependencies: FR3, FR4, FR5, FR6, FR7, R8
Priority: Essential

ID: FR11
Title: Manipulate specific visualization parameters.
Description: Some visualizations have specific parameters that should allow user's manipulation. For each visualization, these specific parameters are:

– Table: none;
– Matrix: nodes weight, grouping, ordering;
- Geographical Referencing: nodes weight, geographical zone;
- Force Directed Graph: Fisheye option, nodes weight, nodes colouring, charge,

distance;
- Circular Graph: nodes weight, nodes colouring, hierarchy (two levels);
- Chord Diagram: grouping.

After alterations in these parameters, the data visualization must be updated to reflect the changes.
Dependencies: FR2
Priority: Essential

ID: FR12
Title: Persistence
Description: All the options made by the user must persist across different data visualizations and must be reset when the index page is visited or a new datasource is selected.
Dependencies: FR1, FR2, FR3, FR4, FR5, FR6
Priority: Desirable

ID: FR13
Title: Visual Feedback
Description: When the application is busy processing (performing a query to the database or drawing a visualization), visual feedback must be provided to the user.
Dependencies: FR7, FR8
Priority: Desirable

5.2.2 Non-functional Requirements

The exploratory Entity Linking application shall fulfil the following non-functional requirements:

ID: NFR1
Title: Data Modifiability
Description: The system should be flexible enough to tolerate changes in the datasources and the parameters under analysis, with minimal effort and without side effects. Those changes must be made in less than a day (8h of work).
Dependencies: None

ID: NFR2
Title: Data Visualizations Modifiability
Description: New data visualizations must be possibly added to the system, without interfering with the remaining visualizations and without affecting the functionalities of the system.
Dependencies: FR2

ID: NFR3
Title: Restrict access to data
Description: Due to data confidentiality constraints, access to datasources must be restricted. Datasources must only be accessed from inside the application.
Dependencies: FR1, DC5

ID: NFR4
Title: Robustness
Description: The application must be fault-tolerant. Events, such as a failed database request, an invalid input served, invalid data, internet connection is lost, etc., must be treated gracefully without creating any sort of impact on the application.
Dependencies: FR1, FR2, FR3, FR4, FR5, FR6, FR7, FR8

ID: NFR5
Title: Reliability
Description: The activity flow and the user actions should remain constant across the entire application, unless a new datasource is selected.
Dependencies: FR13

ID: NFR6
Title: Availability
Description: The average system availability (not considering network failing) must be superior than 98% of the time.
Dependencies: None

ID: NFR7
Title: Usability
Description: A new user, with six months of experience in fraud detection, should become proficient using the app in less than 1 day.
Dependencies: None

5.2.3 Technical Constraints

Technical constraints are technical design decisions which absolutely must be satisfied in the architecture. The system under development must observe the following constraints:

ID: DC1
Title: Web-based
Description: The Entity Linking exploratory tool must be web-based.
Dependencies: None

ID: DC2
Title: REST architectural style
Description: System architecture must follow REST architectural style.
Dependencies: None

ID: DC3
Title: Programming Language
Description: Programming languages must be Java, for the back-end, and JavaScript, for the front-end.
Dependencies: None

ID: DC4
Title: Web Server
Description: Use the free, open source Jetty as web-server.
Dependencies: None

5.2.4 Business Constraints

Business constraints are decisions imposed by business considerations that must be satisfied in the architecture of the system. In these case are:

ID: DC5
Title: Schedule
Description: Product development and validation must be terminated by June 17th, according with the internship legislation, and the thesis delivered by 1st of July. This period might be extended to the 1st of September.
Dependencies: None

ID: DC6
Title: Data Confidentiality
Description: To assure the data confidentiality, the students must sign a confidentiality agreement to have access to datasources containing real data, which are available in a virtual machine only accessible through Feedzai's private network.
Dependencies: None

---

6 Architecture

Software architecture is the centrepiece of a software system design and development. Contains the main design decisions made during the project development and subsequent evolution, defining the structure, behaviour and views of the software system, which provides a solid guideline towards its implementation.

This chapter provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.

In order to depict the software as accurately as possible, the structure of this chapter is based on the "4+1" model view of architecture [46].

6.1 Architectural Representation

This document details the architecture using a series of views, as defined in the "4+1" model: Use Case view, Logical view, Process view, Implementation view and Deployment view.

The Use Case view illustrates the architecture using a small set of case or scenarios that represent some significant, central functionality of the system. The Logical View shows a quick overview of all of the basic sub-systems in the system and gives a basic overview of the system as a whole. The Process View deals with the dynamic aspects of the system, explaining the processes in the system and how they communicate. The Implementation View gives a more in-depth view into how the system has been implemented. The Deployment view describes the mapping of the software onto the hardware and shows how the system is physically configured.

6.2 Use Case View

The Use Case View describes the set of scenarios and/or use cases that represent some significant functionality of the system and have a substantial architectural coverage.

6.2.1 Actors

The actors that interact with the system could be one of these two types:

- **AT01 - User:** has access to the system and can select and alter parameters in order to produce data visualizations.
- **AT02 - System:** the system itself is an actor and handles all the physical and logical processing.

6.2.2 Use Cases

In this project, the focused use cases are:

- **UC01 - Input Selection Parameters**: a User (AT01) interacts with the System (AT02) and introduces a series of parameters to select data to analyse.
- **UC02 – Alter a Selection Parameters**: the User (AT01) can interact with the System (AT02) and alter a selection parameter previously input.
- **UC03 - Manipulate specific visualization parameters**: a User (AT01) interacts with the System (AT02) and is able to manipulate a series of parameters specific to the data visualization.
- **UC04 - Get data from the database:** according with the selection parameters input (UC01), the System (AT02) will get the corresponding data from the database.
- **UC05 - Store data:** when new data from the database is retrieved, the System (AT02) will load the information in memory.
- **UC06 – Get stored Data:** when the User (AT01) alters specific visualizations parameters (UC02), the system will retrieve the stored data to update the visualization (UC07).
- **UC07 - Build a visualization:** after a User (AT01) input, the system uses the data retrieved from the database (UC04) or stored in memory (UC06) to build the visualization.

The diagram representing these use cases is presented in Figure 25:

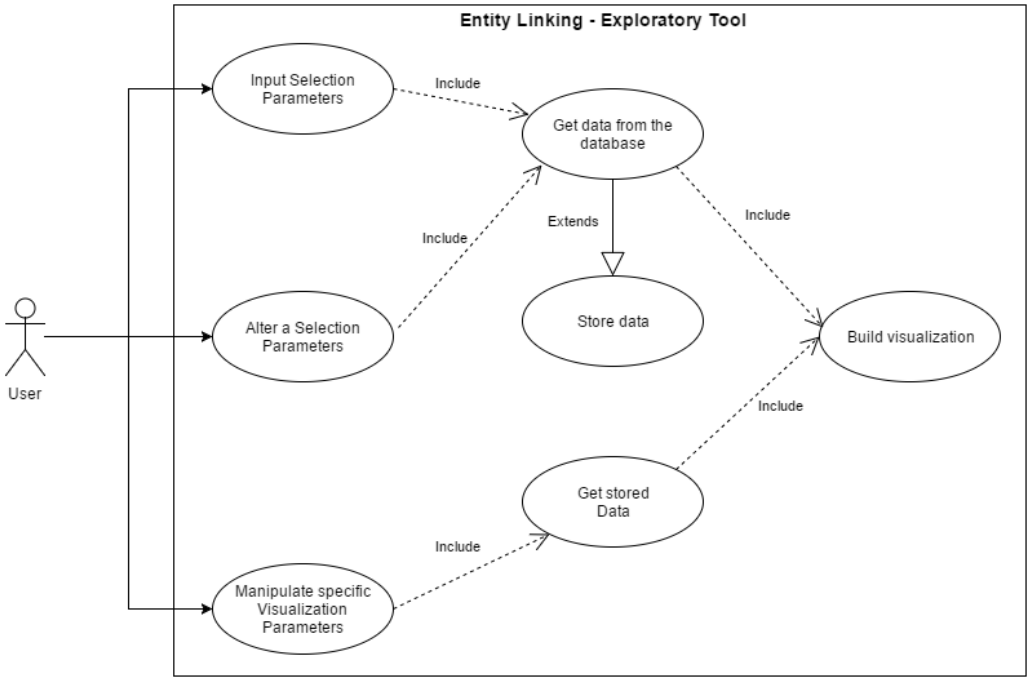

Figure 25: Diagram representing the use cases of the system.

6.2.3 Use Case Realization

The purpose of the use case realization (Figure 26) is to separate the concerns of the system's requirements, represented by use cases, from the concerns of the designers of the system.

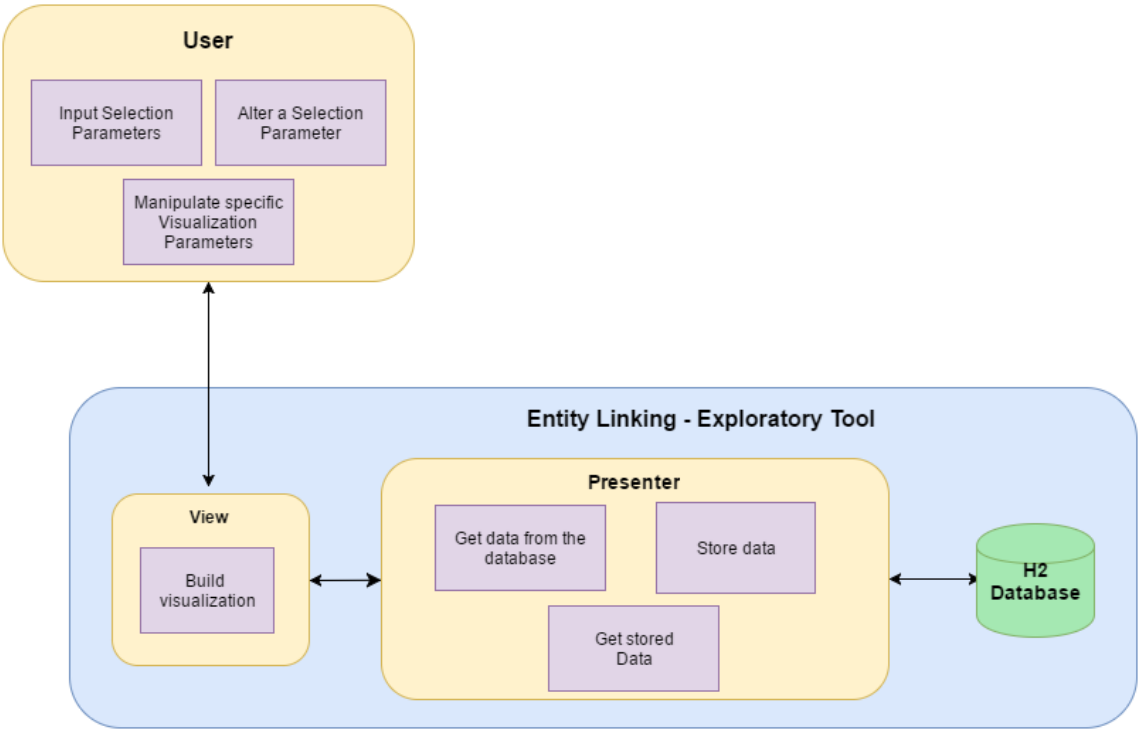

Figure 26: Use-Case realization.

6.3 Logical View

The Logical View describe the architecturally significant logical structure of the system, decomposing it in terms of subsystems, tiers, layers or components.

6.3.1 Client-Server Model

The Entity Linking – Exploratory Tool follows the Client-Server Model.

A Web browser function as a client program, which requests services from a server. In this case, the client establishes a connection to the server over a private network. The services and resources provided by the server are delivered to the client through a Web page.

6.3.2 Multitier Model

The N-tier architecture provides flexibility to an application. The Entity Linking – Exploratory Tool follows a 3-tier architecture, in which presentation, application processing and data management functions are physically separated.

The Presentation tier is the topmost level of the application and can be accessed directly by the user. Communication with the Application Tier is made through HTTP. The Application tier is responsible for the application's functionality and processing. It communicates with the Presentation tier though HTTP and with the Data tier through a JDBC connection.

The Data tier is responsible for the data storage mechanisms and database access. A JDBC connection is used to connect with the Application Tier.

6.3.3 Layered Model

A multilayer architecture approach groups different responsibilities in different logical layers. This web application is divided into three layers: Presentation layer, Application layer and Data layer.

The Presentation layer contains the web pages and handles all the input from and all the output to the user. The Application layer handles the business logic and provides an abstraction to the database. The Data layer encapsulates the database.

6.3.4 Model-View-Presenter Pattern

The Entity Linking – Exploratory Tool web uses the Model-View-Presenter architectural pattern [68] to build user interfaces. This pattern divides the software application in three interconnected components, separating the model from the view and from the presenter.

The model interface manages data directly in the H2 Database. The View is a passive interface that displays the data outputted by the Presenter and routes user inputs to the Presenter to act upon that data. The Presenter is the central component of the MVP architecture. It acts upon the model and the view, receiving inputs from those components, converting those inputs to commands, retrieving data from repositories in the model and formatting data to be displayed on the view. This project uses Jetty, a Java HTTP server and Java Servlet container, as the application server, at the core of the Presenter component. Communication with the view is made through REST endpoints and communication with the model uses a JDBC connection.

6.3.5 Architectural Dependencies

Figure 27 represents the architectural dependencies between the subsystems, tiers, layers and components previously considered.

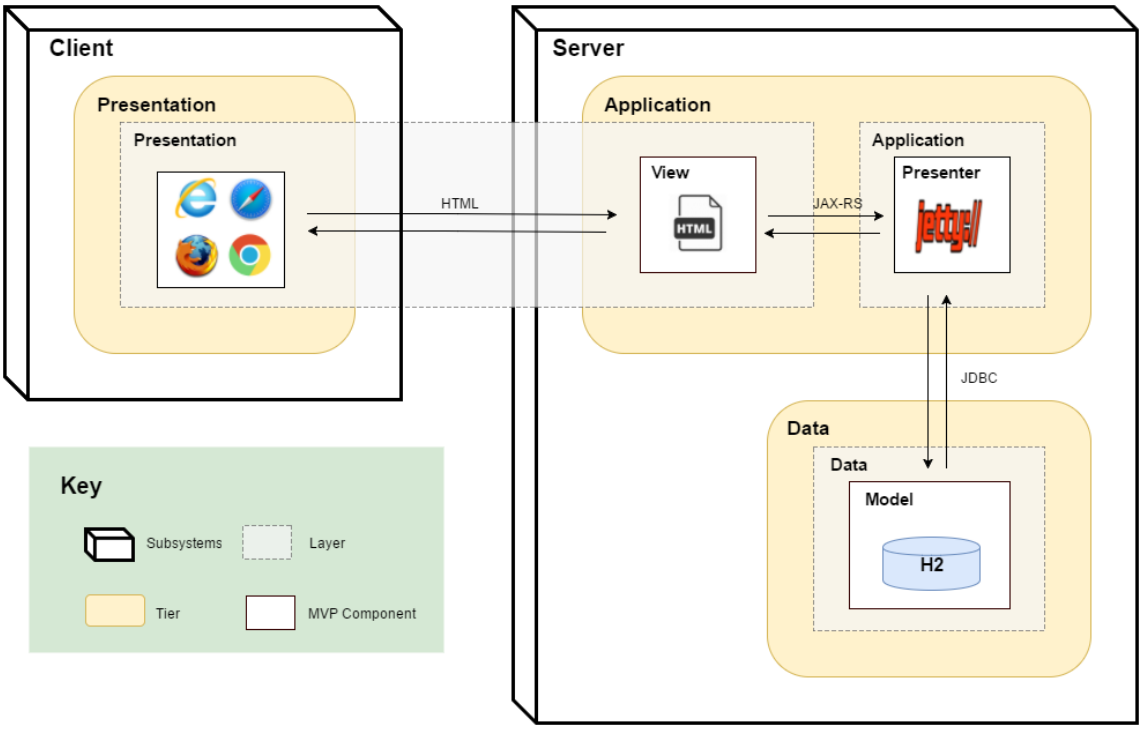

Figure 27: Logical View representation of the system.

6.4 Process View

The process view deals with the dynamics aspects of the system by defining the system's processes and their communication. Therefore, focuses on the system's runtime behaviour.

It is defined as a process a set of tasks that form an executable unit. Each task represents a separate thread of control that can be scheduled individually. Processes represent the level at which the process architecture can be intentionally controlled.

To provide a basic understanding of Entity Linking - Exploratory Tool process organization, figure 28 models the flow of the process of analyzing data through the available data visualizations on the system.

6.5 Implementation View

The implementation view reflects the inner organization of the software development environment. Software application is divided in subsystems, or packages, organized in a hierarchy of layers, in which each layer provides a well-defined interface to the others.

The packages that make up the Entity Linking - Exploratory Tool, and their dependencies, are depicted in the package diagram on figure 29. This application is constituted by four packages: Presentation, Server, Database and Resources.

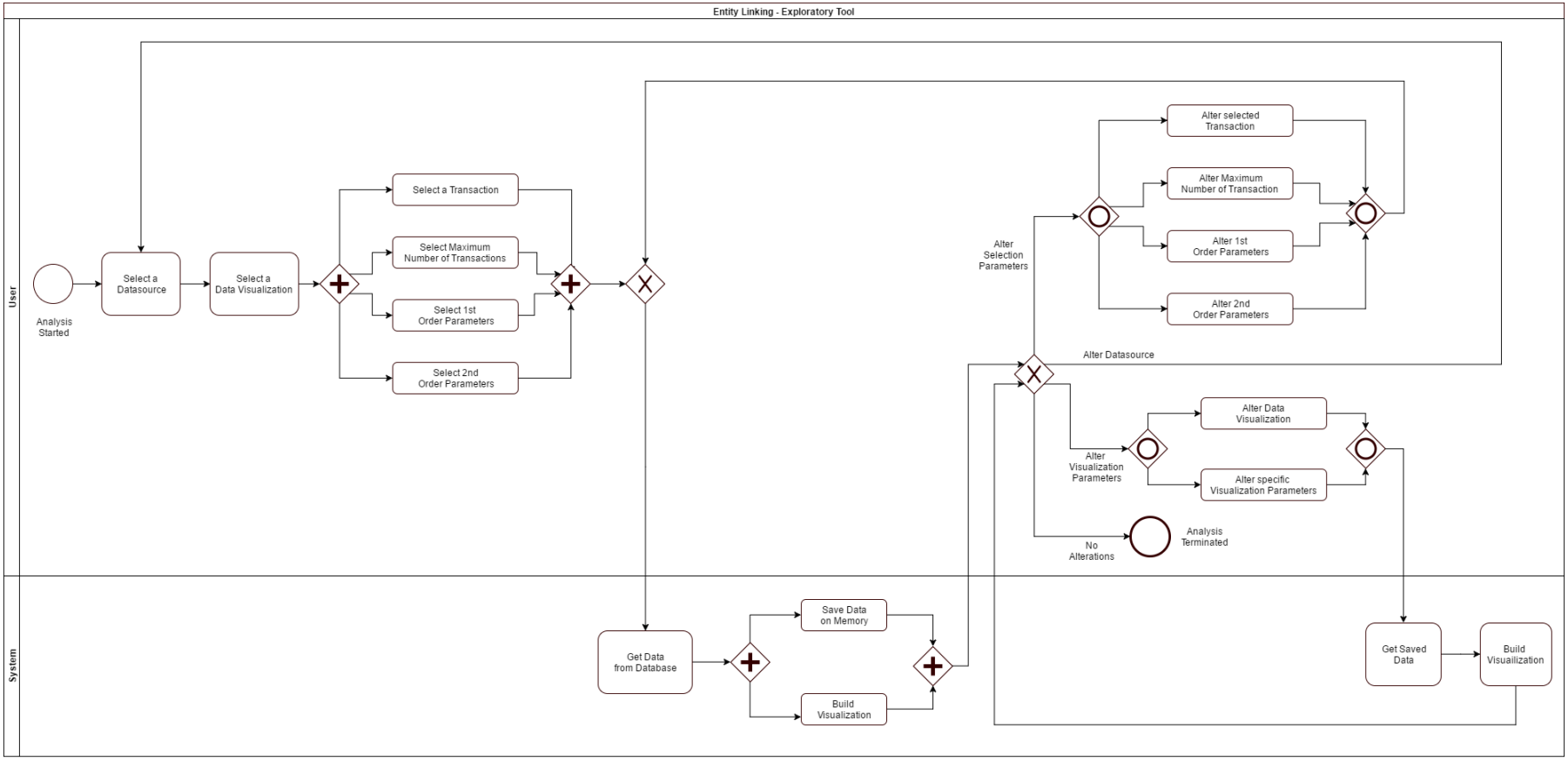

Figure 28: Process modelling of Entity Linking - Exploratory Tool using BPMN.

Presentation package is concerned with preparing and present the data in a human-readable form. The server is responsible by processing and manipulating collected data. Database package handles the data storage. The resources contains generic representations of data, to be used by the server and database packages. Relationships between these packages are depicted on the Package diagram.

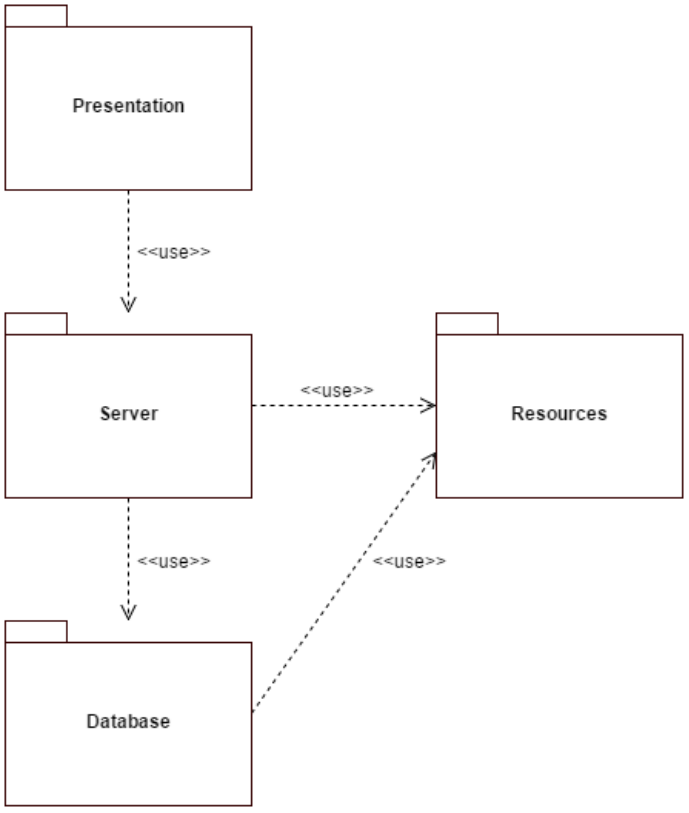

Figure 29: Entity Linking - Exploratory Tool package diagram.

6.6 Physical View

The physical view represents the topology of software components of the system on the physical layer, as well as their connections.

The deployment diagram, featured on figure 30 model the hardware components from the Entity Linking - Exploratory Tool (Web Client, Application Server and Database Server), which software components run on each node (e.g., web application, database) and how the different components are connected (in these case, HTTP, JAX-RS and JDBC).

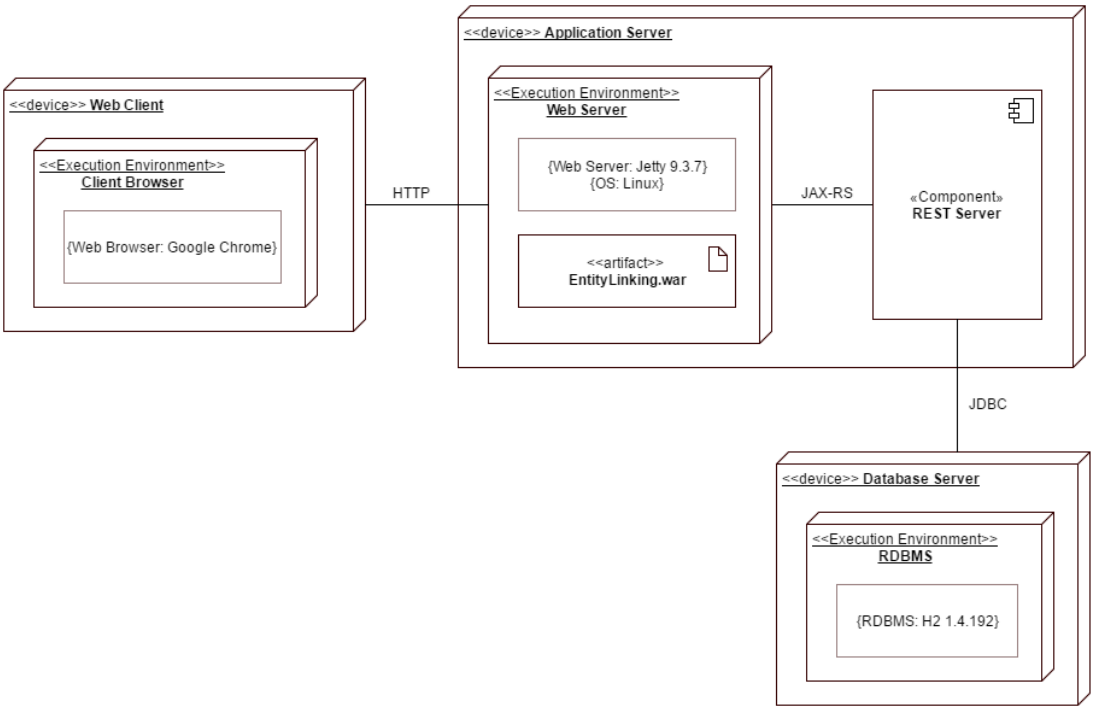

Figure 30: Entity Linking - Exploratory Tool deployment diagram.

7 Risk Management

A risk is an event or condition that, if it happens, can have a positive or negative effect on the project. The process of identifying, assessing, responding, monitoring and reporting risks is designated as risk management. This chapter address the task of risk management on this project, focusing on the risk management procedure and the risk log.

7.1 Risk Management procedure

Risk Management procedure defines how the risks associated with this project will be identified, analysed and manged, outlining the activities performed throughout the lifecycle of this project.

7.1.1 Process

Throughout the life of the project, risks will be actively identified, analysed and managed. Identification of the risks will be made as early as possible in the project, to minimize their impact.

Identified risks will be collected in a risk management log.

7.1.2 Risk Analysis

All risks identifies will be classified according with their probability of occurrence and impact to the project. This classification will determine which risks are the top risks to pursue and respond, and which risks can be ignored.

The probability of occurrence of a risk can be classified as:

- High: Greater than 70% probability of occurrence
- Medium: Between 30% and 70% probability of occurrence
- Low: Below 30% probability of occurrence

The overall impact of a risk will be defined as:

- High: Risk that has the potential to greatly impact project cost, project schedule or performance
- Medium: Risk that has the potential to slightly impact project cost, project schedule or performance
- Low: Risk that has relatively little impact on cost, schedule or performance

According with the scale from figure 31, risks that fall within the red and yellow zones will have risk response planning which may include both a risk mitigation and a risk contingency plan.

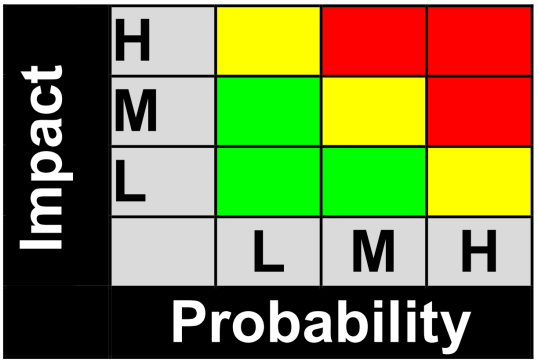

Figure 31: Scale to classify risks according with their probability of occurrence and overall impact to the project.

7.1.3 Risk Response Plan

For each major risk, one of the following approaches will be selected to address it:

- Avoid: eliminating the cause of the problem, to eliminate the treat;
- Mitigate: identify ways to reduce the probability or the impact of the risk;
- Accept: nothing will be done;

7.1.4 Risk Monitoring

Risk analysis will be performed throughout the lifecycle of the project and a risk log with all identified risks will be maintained.

7.2 Risk Log

Risks identified throughout the lifecycle of this project are summarized on figure 32.

ID	Risk	Mitigation	Contingency	Probability	Impact
1	Timeframe: the project is not finished on time		Deadline can be postponed for the September special period evaluation.	M	H
2	Datastorage limitations: the use of an embedded database might limit the capacity of store and process data	Use a different database system	Reduce the scope of the application	M	M
3	Data confidentiality: only authorized personnel can access the data which limits the availability	Perform tasks, like the usability tests, with authorized personnel only	Anonymise data to use outside the company	H	H
4	Use of virtual machine: confidential data are kept in a virtual machine only accessible through Feedzai's private network which slows down the development process		Anonymise data to use outside the company	H	H
5	Dataset inconstancies demo dataset was automatically generated and doesn't reflect real data patterns, nor have the same parameters which can impact system's development	Test the application with real datasets, as much as possible	Create a generic application, as much as possible, which can be adapted to different datasets.	H	H
6	Display capacity: the amount of information encoded in a data visualization is limited	Limit the amount of data to be displayed on each visualization	Change the ambit of the visualization in order to display group transaction instead of single elements	H	H
7	Time: Underestimation of the time required to get acquainted with D3 library		Readjust timeframe	M	M
8	Colour scale: D3 scales have a 20 colours limit and some shades are similar		Create a personalized scale	M	L
9	Effectiveness of the visualization: data visualization is not good in detecting fraud	Try to explore different options within the visualization	Select a different data visualization	M	H
10	Validation: validating the effectiveness of a data visualization is difficult because there are too many possible questions on the table		Define a set of parameters to be analysed, in order to find the very best choice according to them.	H	M
11	Usability test: difficulties in gathering an heterogeneous group of test volunteers		Perform the test with a more homogeneous, group, but as heterogeneous as possible	M	M

Figure 32: Risk log for Entity Linking project.

8 Implementation

This chapter describes how the system was implemented. First, the technologies used on this project are listed. Afterwards, a deeper description of the system's components is presented. At last, we delve into the employed data visualizations, explaining their purpose, functionality and implementation details.

A system's walkthrough, in the user perspective, is provided in Appendix B. This guide intents to clarify how the user interacts with the system and how it can be used to detect fraud.

8.1 Technologies

This section introduces the technologies adopted in the development of this project, focusing on the libraries used on front-end development.

8.1.1 Back-End Technologies

Several programming languages, libraries and frameworks were used on the development of this project. Table 2 summarize technologies used for Back-End.

Technologies such as Maven, H2, JDBC and Jersey were chosen to maintain coherence with the Alert Manager application. The use of Java and Jetty come from the technical constraints of this project (DC2 and DC3, from section 5.2.3).

Apache Maven was chosen as the project repository, which is used to hold build artifacts and dependencies of varying types. H2 Database engine is a very fast database engine, which supports SQL and JDBC API. Therefore, a JDBC connection is used to communicate with the H2 Database. Jetty is used as the application web server and servlet container. Jersey framework was chosen as JAX-RS Reference Implementation, used to communicate with the Front-End. The programming language used was Java.

Technology	Description
Java (JDK8)	General-purpose computer programming language
Jersey 1.19	JAX-RS Reference Implementation framework, which provides support in creating web services according to the REST architectural pattern
JDBC 4.2	Java API, defining database access
Jetty 9.3.7	Java HTTP (Web) server and Java Servlet container, developed as a free and open source project by the Eclipse Foundation
H2 1.4.191	Embedded Java SQL database
Maven 1.7	Apache Maven is a software project management and build automation tool, used primarily for Java projects

Table 2: Back-end technologies used on this project.

8.1.2 Front-End Technologies

Since the Front-End is the main focus of the application, there was more freedom of choice to select Front-End Technologies. Technologies used for Front-End development are summarized on table 3.

The use of JavaScript as front-end programming language come from the technical constraints of this project (DC3, from section 5.2.3). HTML and CSS are two of the core technologies for building web pages. Bootstrap API was chosen as UI elements toolkit, due to its comprehensive set of UI elements, popularity and outstanding community. To build data visualizations, D3 library was chosen, due to its powerful visualization components and data-driven approach to DOM manipulation. This library uses SVG.

Technology	Description
HTML 5.0	Standard markup language for creating web pages and web applications
CSS	Style sheet language used for describing the presentation of a document written in a markup language.
SVG 1.1	XML-based vector image format for two-dimensional graphics with support for interactivity and animation
JavaScript	High-level, dynamic, untyped, and interpreted programming language
jQuery 1.9	Friendly and well documented cross-platform JavaScript library, gives the ability to perform AJAX requests and easily manipulate the DOM
Bootstrap 3.3.7	Free and open-source front-end web framework, which provides several ready-to-use customizable UI elements
D3.js 4.2.2	JavaScript library for manipulating documents based on data, using HTML, SVG, and CSS

Table 3: Front-end technologies adopted on the development of this project.

8.1.3 jQuery

jQuery[3] is a cross-platform lightweight JavaScript library, used for client-side web development. The syntax of jQuery is designed to simplify HTML document traversal and manipulation, event handling and the development of AJAX applications.

8.1.4 D3.js

D3.js[2], or D3, from Data-Driven Documents, is a JavaScript library used to manipulate documents based on data. Its primary function is to create data visualizations, operating over HTML, SVG and CSS with pre-built JavaScript functions.

The main functionalities offered by this library are:

- Loading data into memory;
- Binding large datasets to SVG elements within the document, creating new elements as needed;
- Transforming those elements, mapping each element's bound datum and settings to its visual properties;
- Transitioning elements between states in response to user manipulation and input.

8.1.5 Bootstrap

Bootstrap[1] is a cross-platform web framework, used to design responsive, mobile-first websites and web applications. It includes HTML and CSS based design templates for common user interface components, enabling the easy creation of responsive layouts.

8.2 Components

This section describes the main components of the system: the database server, the web server and the web client.

8.2.1 Database Server

The database server is responsible by accessing and manipulating data. To do that, a H2 database engine is used. H2 is a Java SQL database which uses JDBC API to establish connections. On this project, H2 embedded mode was used.

According with the described functional requirements (see FR1 in section 5.2.1), the web application Entity Linking must be able to interact with several datasets. The nature, structure and properties of those datasets may vary, and the backend must have the versatility to accommodate those variations.

This is achieved through Inheritance. The main database class functions as a superclass and each dataset has a corresponding class that function as a subclass of the parent database, inheriting its attributes and methods. Each subclass overrides the necessary methods to function.

Three datasets are currently integrated with the application: a demo dataset, automatically generated, and two datasets with real commercial data. Due the sensitive nature of the data, these datasets are only accessible through Feedzai's private network and only by authorized personnel.

Each dataset contains a table with commercial transactions, with tens of attributes (similar to those described in Feedzai's developer documentation [34], represented in figure 33). These attributes vary from dataset to dataset and the data collected from

each data source can vary. Therefore, the parameters withdrawn from a dataset are specific to that dataset and are declared in each database subclass. Those parameters were selected by the development team, considering their relevance to identify entities, and were approved by the project stakeholders.

The size of each dataset differs significantly, with the demo database ranging 2k records, the first sample 200k records and the second sample 1500k records. This size difference proved to be quite challenging, especially since the application was developed based on the demo dataset. When the sample datasets were integrated on the application, code refactoring was necessary to improve performance.

A depth-first search algorithm was implemented, to search for relationships between data entries. This algorithm searches for first-order relationships between the selected transaction and the dataset. Then, it searches for second-order and above relationships between result set and the data set until a maximum number of relations, defined by the user in the web application, is reached or until no more relations can be found in the dataset.

Name	Description
`user_id`* string	The unique user ID for the buyer, used in your system. Example: *"af00-bc14-1245"*
`amount`* integer	The payment's amount in cents. Example: *11099* (given the currency of *"USD"*, represents $110.99). If the currency is not specified in the *"currency"* field, the merchant's default currency will be assumed
`ip` string	The user device IP at purchase time. Example: *"89.180.212.63"*
`currency` string	The payment's currency (3-letter ISO 4217 currency code). Example: *"USD"*, *"EUR"*. If not specified, the merchant's default currency will be used
`id` string	The unique payment ID in your system. If omitted, we automatically generate one. Example: *"124212-00245"*
`items` array	The list of items that were payed for
`transaction_type` string	The type of the transaction, according to your own business. Examples: *"sale"*, a money *"transfer"*, a *"return"*. In case of sending post-authorization, you should use the *"auth"* transaction type
`payment_method` string	The payment method used. Examples: *"card"*, *"voucher"*, *"cash"*, *"paypal"*
`card_cvv_present` boolean	Indicates if the card's CVV was provided/checked when accepting card information and the CVV code was valid. If the CVV was not checked or it failed validation this should be false
`user_email` string	The user's registered Email address. Examples: *"howey.1975@gmail.com"*
`user_fullname` string	The user's full name, as registered in your system. Examples: *"John Smith"*, *"Joanne Albert Doe"*
`user_created_at` long	The date the user first registered or appeared in your site (in milliseconds since the Unix Epoch, UTC timezone). If omitted, we assume the request time. Example: *1368457861425* (represents May 13 16:13:57 WEST 2013)
`user_gender` string	*"M"* when the user is Male, *"F"* when the user is female, *"O"* in other/undefined cases
`user_dateofbirth` string	The user's date of birth in the format YYYY/MM/DD. Example: *"1975/06/30"*

Figure 33: Exemplification of some of the attributes found in a transaction object, according with Feedzai's developer documentation [34].

8.2.2 Web Server

The web server functions as a bridge between the web client and the database server. It receives inputs from the web client through HTTP, processes them and updates the view accordingly.

Communication with the database server is made through JAX-RS. JAX-RS is a Java programming language API for RESTful Services. This framework uses annotations to map a resource class (a POJO) as a web resource.

To persist user choices across the platform (FR12 from section 5.2.1), the value of parameters manipulated by the user is saved in memory as session parameters, and updated every time the user alters them. When no user input is available, those parameters are initialized with default values.

To minimize requests made to database, data returned from the database server is saved in memory as a session parameter. This data is accessed to build different visualizations over the same parameters or to refresh a visualization when specific parameters are altered that don't affect the selection of transactions on display.

8.2.3 Web Client

The web client is responsible by user interaction. This component features a highly interactive dashboard, accessible over a web browser. The user interacts with it to select and manipulate parameters that will determine the data visualization displayed on the browser.

When the application is busy processing a user request, visual feedback is provided to the user (as in accordance with FR13, on section 5.2.1) to inform that a background operation is being performed.

8.3 Data Visualizations

This section addresses data visualizations implemented on the application: table, co-occurrence matrix, geographical referencing, force directed graph, circular diagram and chord diagram.

8.3.1 Table

A table visualization displays data, in its raw numerical form, using a tabular view. Due to the quantity of information displayed on a table, evaluating the information is hard and data patterns are almost impossible to grasp. Notwithstanding the limitations of a table, the widespread familiarity of the public with this type of data visualization makes it a good starting point to explore relationships between data.

In the Entity Linking application, a table visualization was built using HTML, CSS and Bootstrap. Selected transaction is displayed on top of the table, highlighted in

purple. Related transactions are listed below and the fields related with the selected transaction, according with the parameters input by the user, are highlighted in light purple (see figure 34).

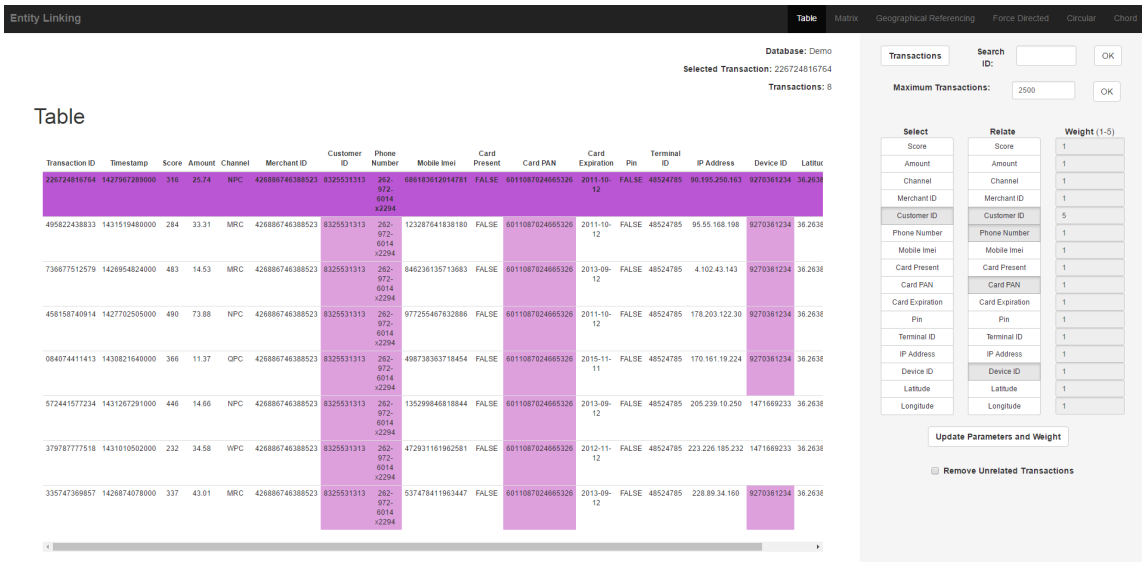

Figure 34: Table view representing a list of transactions with the same Client ID, evidencing relations by Customer ID, Phone Number, Card PAN and Device ID.

This table has no interactivity with the user. Some interactive scenarios were tested, such as altering the selected transaction by selecting a row of the table, or order the transactions by their relationship proximity with the selected transaction, but proved to be too disruptive for the application flow and were not integrated in the final version.

8.3.2 Matrix Diagram

Relationships between transactions can be represented as a matrix diagram. This adjacency matrix is a square symmetric matrix in which each row and column represent a transaction. The elements of the matrix indicate whether the respective transactions are related or not and the colour gradation encodes the force of that relation, in accordance with the relation parameters input by the user. Selected transaction can be distinguished from the others though an asterisk, marking its position on the grid.

A "Group By" parameter can be defined through a drop-down menu (figure 35), which groups transactions according with it by colour. Related transaction with the same value on that parameter will be painted with different gradations of the same colour. When not related, they will be coloured in a grey scale. Different colours were achieved using D3 built in colour palette, the d3 category20. Different shades of the same colour were obtained through different levels of opacity, using a linear scale with clamping.

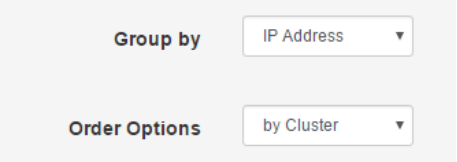

Figure 35: Specific options of the matrix visualization.

The effectiveness of a matrix diagram is heavily dependent on the order of its rows and columns. When related nodes are placed closed together, it becomes easier to identify clusters and bridges. Therefore, different "Order By" options are offered in this visualization, through a drop-down menu: name, frequency and cluster (figure 35). The "Name" option orders the transactions alphabetically. "Frequency" orders the transactions according with the number of established relationships. "Cluster" groups transactions with the same value of the "Group By" parameter.

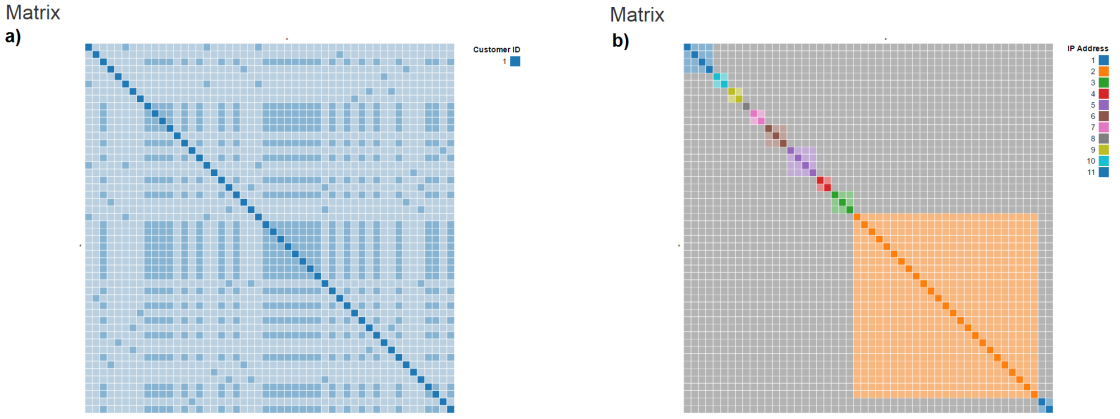

Figure 36: Co-occurrence matrix representing 50 transactions originated from the same Customer, grouped by cluster.

Matrix a) is coloured by Customer ID, evidencing that all transactions were originated from the same customer and matrix b) is coloured by IP address. The 50 transactions from the same customer ID were made from 11 different IP addresses.

A tooltip is attached to each cell. When the user hovers the cursor over a cell, a hover box appears containing the similarities and the differences between the two transactions represented by that cell.

8.3.3 Geographical Referencing

The purpose of the Geographical Referencing data visualization is to represent geo-localized transactions, based on the latitude and longitude coordinates from where the transaction was originated.

To render a map, D3 and TopoJSON were used. TopoJSON [19] is an extension of GeoJSON, which encodes topology. TopoJSON eliminates redundancy on its files by stiching geometries together with shared line segments, called arcs.

Geographic data files were collected from the internet, after a thorough research. Geojson-regions open-source project [48] permits the build of GeoJSON customized maps, based on Natural Earth public domain data [4]. Several tools are available on the internet to convert GeoJSON files in TopoJSON. In this project, geojson-topojson open-source web-interface was used [63].

To render geography with D3 library, two more things are needed: a projection and a path generator. The projection defines how to project spherical coordinates in the Cartesian plan, which is needed to convert 3D coordinates to2D. The path generator formats the projected 2D geometry appropriately in the SVG element, where the visualization will be built. In this project, a Mercator projection was used, in which the world is projected to a rectangle.

Using geo position information, available in the dataset, each transaction was displayed as a point in the map. Links between transactions represent relationships between them, being the relationship strength encoded by the thickness of the link. However, this proved to be cumbersome and redundant, since several transactions can be made from the same location. Therefore, we opt to group transactions by location and represent the ligations between those grouped transactions (see figure 37). A tooltip indicates on each node the ID and the Customer ID of each transaction grouped there.

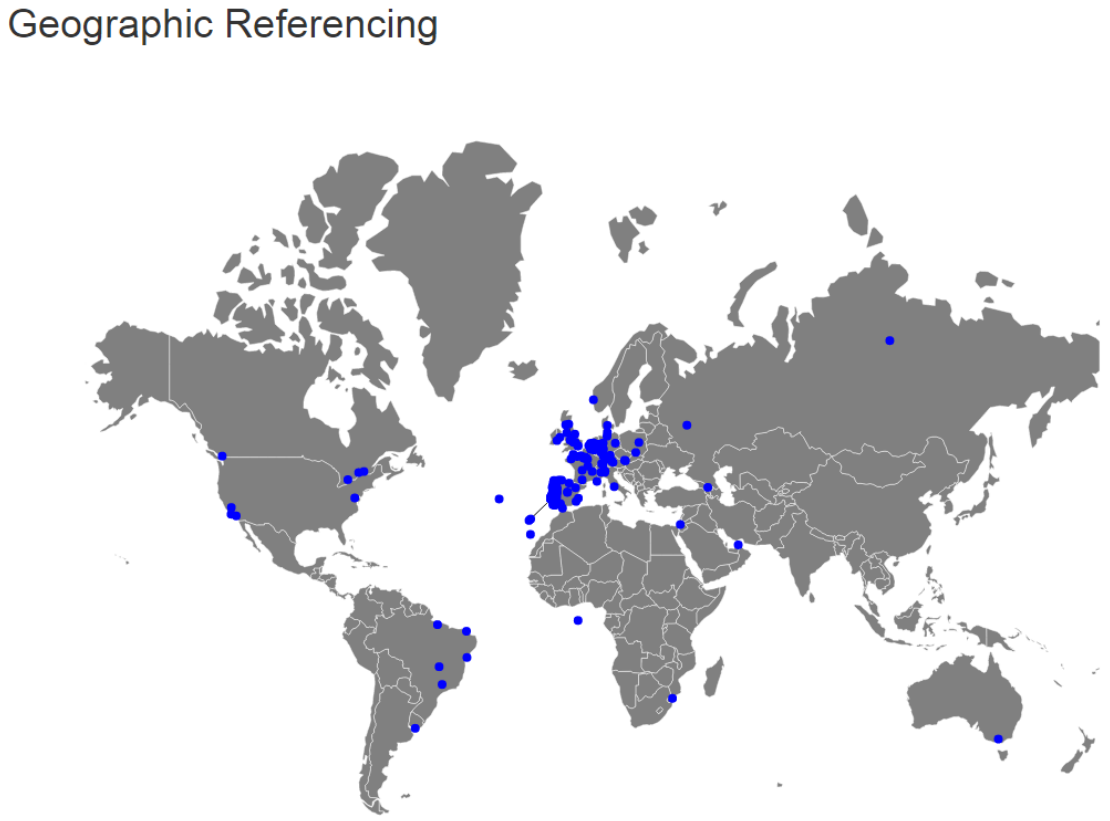

Figure 37: Geographical representation of relationships between Customer ID from transactions made to the same Merchant.

Ideally, it should be possible to fully explore the map to have a better grasp of the information displayed. Functions such as zoom and pan are quite easy to implement in the map. However, several problems arose while projecting the nodes and the links over the new maps coordinates. Another approach was to apply fisheye distortion to the visualization, in order to magnify the local region around the mouse cursor, while leaving the rest of the map unaffected. Although this distortion was easily applied to the graph, it didn't function well over the map. As a last resort, we opt to provide several maps, corresponding to different regions of the globe, and the user chooses which region must be displayed through a drop-down menu (figure 38).

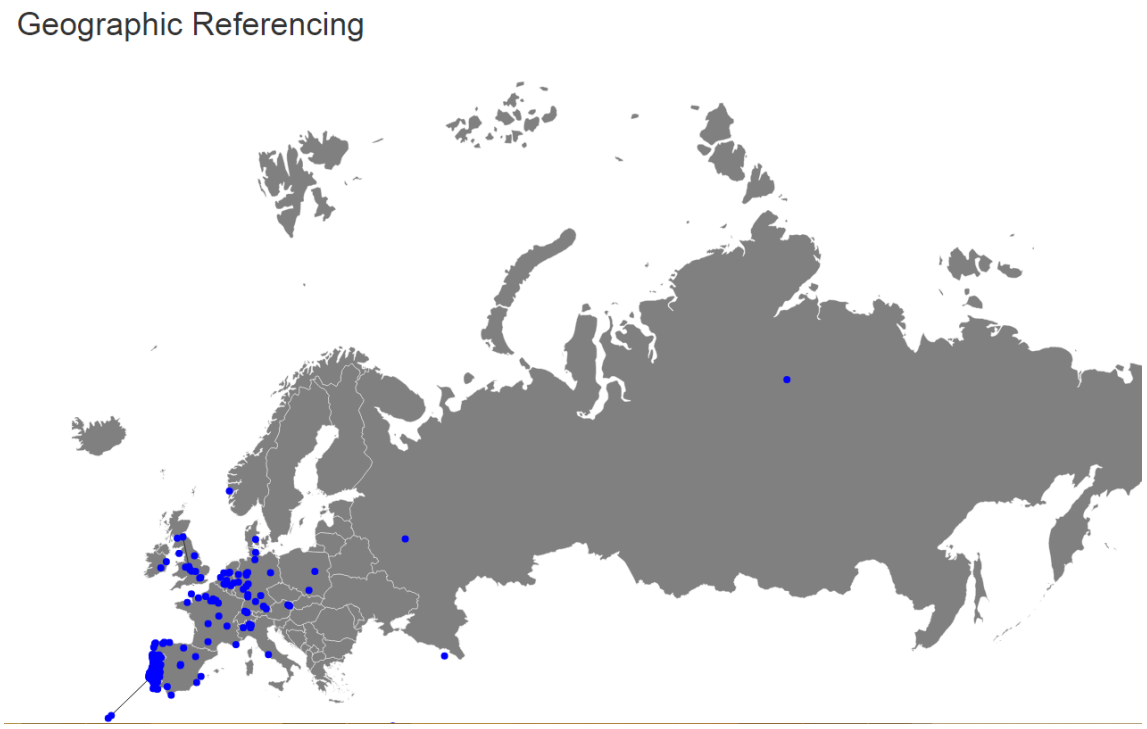

Figure 38: Europe focus of the data represented in figure 37.

8.3.4 Force Directed Graph

Force-directed methods are used to draw graphs in an aesthetically pleasing way. These algorithms combine attractive forces on adjacent vertices with repulsive forces on all vertices, and use those forces to simulate the motion of edges and nodes until the system is able to minimize their energy and stabilize.

D3 library has a module dedicated to Force Directed Layouts, which implements a velocity Verlet numerical integrator on particles, to simulate physical forces. Force Atlas [43] is the default layout algorithm: while it runs, the nodes repulse each other and the edges attract the nodes. This is done through tick events, which updates the velocity and position according with the force of the node. This algorithm has advantages and disadvantages: it's easy to implement but the distribution of the nodes depend on the other nodes and depend on its initial state. Therefore, this algorithm can get stuck in a local minimum.

Force is applied continuously, as long as the layout is running. This brings some difficulties to properly analyse the data contained in the visualization. Therefore, an alternative static layout was tested. Instead of updating the graph with each tick, the graph is run a fixed number of times and displayed once. This was the version implemented on the Entity Linking application.

To better observe micro and macro features simultaneously, fisheye distortion was applied to the visualization. User can opt to add this feature by checking the "Fish-eye" checkbox on the right menu. This distortion magnifies the region circularly around the mouse cursor, while leaving the rest of the graph unaffected.

Two parameters can be modified by the user: charge and distance. The charge modifies nodes positions and velocities, acting as a physical force similar to an electrical charge or to gravity. The distance correspond to the maximum distance between linked nodes. The link force pushes linked nodes together proportionally to the strength of the relationship. Therefore, the distance between linked nodes will be smaller as more related they are.

Several options were tested to figure out the best way to represent the links between nodes. The strength of the relationship is encoded at two levels: by the distance between nodes, which decreases with the strength, and by the thickness of the line representing the link, which increases with the relationship strength.

Different types of D3 scales were tested to encode those relationships: linear, exponential and logarithmic. An exponential scale makes smaller differences more noticeable, while differences between nodes highly related are mostly undetected. A logarithmic scale has the opposite effect, making less related nodes almost undistinguishable and differences between nodes highly related are more noticeable. Since Entity Linking is an exploratory tool and the intention of the analysis can vary, we opt by using the most neutral option, the linear scale.

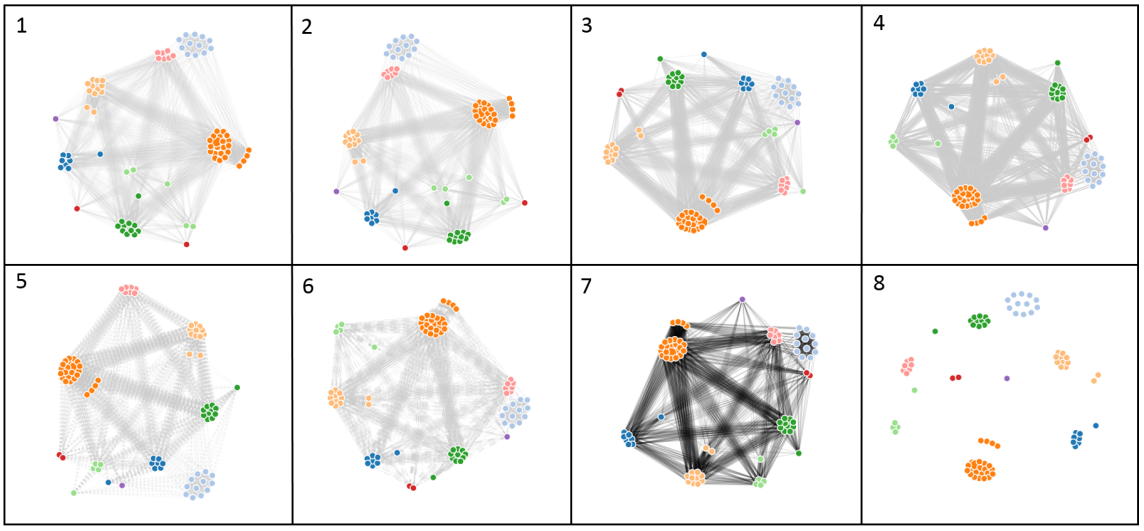

Figure 39: Force Directed representation with different opacity (1 to 3), thickness (4), style (5 and 6) and colour (7 and 8) line options.

In complex highly connected graphs, the ligations can become a visually distraction. To minimize that effect, different colour, thickness, opacity and style parameters were tested (figure 39).

Colour encoding is used to differentiate nodes by a parameter value. The default parameter is the Customer identification. However, a drop-down menu allows the user to select other parameter to colour the nodes. Different colours were achieved using the d3 category20 built in colour palette.

Selected transaction can be distinguished from the other by the size of its node, which is bigger than the rest. Each node has a tooltip associated which displays all the information related with the transaction (figure 40).

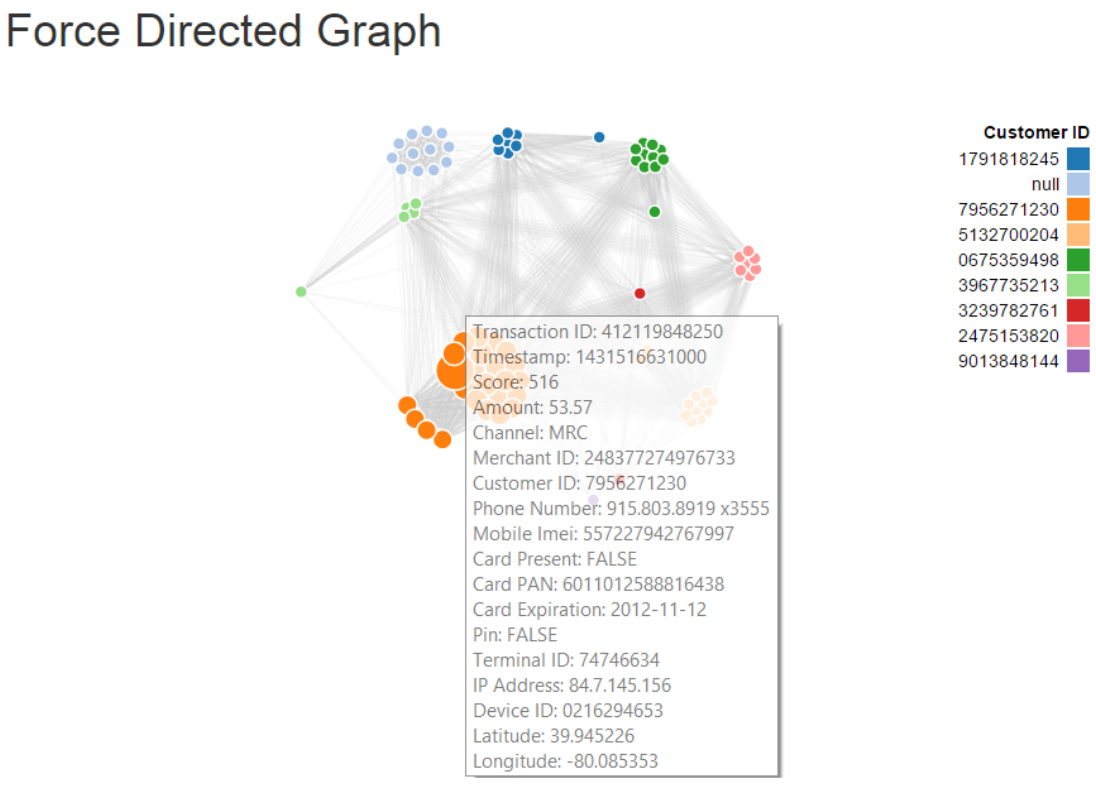

Figure 40: Force Directed Graph.

Other implementations options were analysed, to improve the interactivity of the graph. Those were: collision detection; highlight selected nodes; collapsing nodes; fixing nodes by pinning them down. To improve graph performance, we consider implement the algorithm of Yifan Hu [42], which is more efficient than the Force Atlas used. However, these options were not implemented in the final version of the force directed graph, contained in the Entity Linking application.

8.3.5 Circular Diagram

In a circular layout, nodes are represented around a circumference. Relationships between nodes are represented as ligations. This type of layout supports edge bundling. Using edge bundling significantly reduces the visual clutter in drawing a graph with a large number of edges.

In this implementation, we used a hierarchical edge bundling algorithm and two layouts: a radial d3.layout.cluster to position the tree nodes, and d3.layout.bundle to group the dependencies into spline bundles. Dependencies are bundled according with two parameters, selected by the user through drop-down menus (figure 41). Second level hierarchy option is only available after a first level hierarchy parameter has been chosen. Once a parameter is chosen in one of the drop-downs menu, it does not appear in the other.

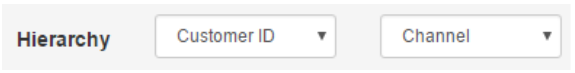

Figure 41: Drop-down menus to input hierarchical options for circular diagram.

Colour encoding is used to differentiate nodes by a parameter value, input by the user. Different colours were achieved using the d3 category20 built in colour palette.

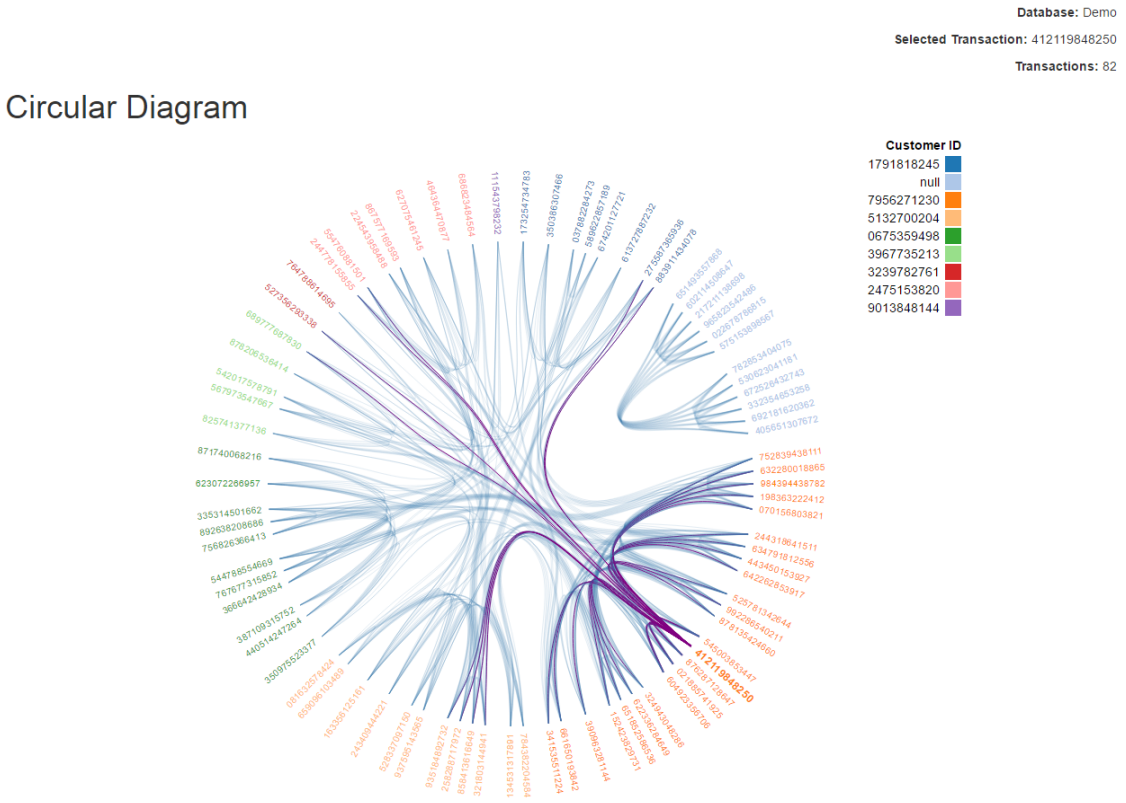

Figure 42: Circular diagram, representing transactions with the same Merchant ID, related by Customer ID and Channel, bundled by Customer ID (first level hierarchy parameter) and Channel (second level hierarchy parameter).

The ID of selected transaction is displayed as bold and with a font size bigger than the others. Relationships established by the selected transaction are marked with a purple link, to be distinguished from the other (figure 42).

Mouseover any of the nodes in the network and the relationships established are marked in green. Each node has a tooltip associated which displays all the information related with the transaction.

8.3.6 Chord Diagram

A chord diagram is a graphical method of displaying inter-relationships between data in a matrix. It is based on the circular diagram, but the relationships are established among a group of entities.

In the chord layout (figure 43), a ribbon represents two cell, except for cells on the diagonal, which are represented by a single ribbon. The thickness of the ribbon represents the magnitude of the relationship between elements.

Through a drop-down menu, users can select the parameter to be used to group transactions.

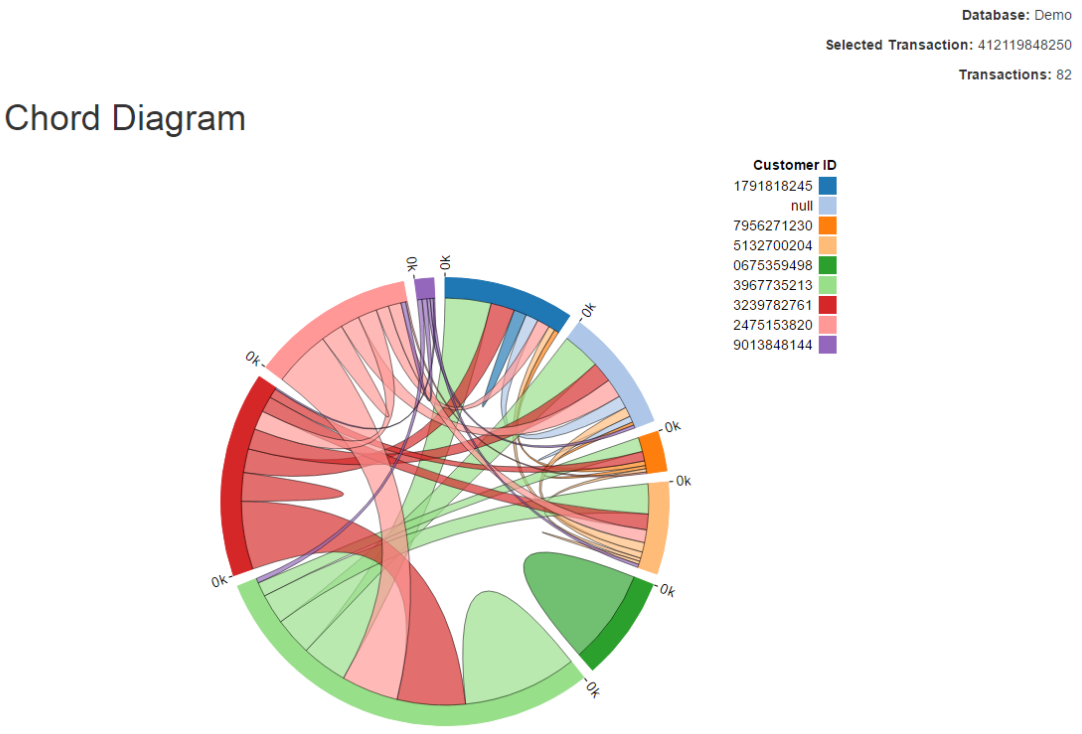

Figure 43: Chord diagram representing transactions with the same Merchant ID, related by Channel and grouped by Customer ID.

9 Validation

In any software project, testing is a fundamental step of the development process [15]. Through testing, it is possible to validate whether the developed system meets all the requirements and obey all the constraints and to validate whether the components behave as expected. Furthermore, it can be used for quality assurance.

Testing encompass several activities and techniques, depending on the objectives of the test. In this project, Usability Testing will be used to evaluate the Data Visualizations developed for Entity Linking.

This chapter covers this validations step performed on the system. First, an introduction to usability evaluation is provided. The following section introduces the testing plan and, after that, the results are presented. Last sections comprised a set of suggestions and recommendations to improve the usability of the system under test.

9.1 Usability Evaluation

This section function as a brief introduction to Usability Evaluation. First, the factors analysed with this type are introduces. Then, some of the methodologies used are presented. At last, we will look into the participants in the usability test.

9.1.1 Usability Factores

Several usability factors can be analysed in a Usability Evaluation [61] including:

- Easy of learning: promptitude with which a new user is able to interact with the user interface.
- Efficiency of use: promptitude with which an experienced user can accomplish tasks.
- Memorability: capacity of a user to recollect previous experiencing with the application in order to use it effectively in subsequent visits.
- Robustness: quantity and severity of error made by subjects while using the platform and how easy is to recover from those errors.
- Intuitive design: how effortless is to understand and navigate through the application.
- Subjective satisfaction: satisfaction level of the subject with the usage experiment.

9.1.2 Methodology

Several methodologies can be used to perform usability tests.

Usually, this kind of testing requires the creation of a scenario or a realistic situation where the test users are to perform a set of representative tasks of the system. Those tasks are to be performed individually and autonomously by the test subjects, without interference by the experimenter, who must keep an observer role in the process.

Diverse test instruments can be used, from scripted instructions and paper prototypes to test questionnaires, in order to gather feedback from the user.

9.1.3 Participants

Usability tests have associated costs and, although efficiency increases with the number of tests performed, the informational gain not always compensate the cost of additional testing.

The number of tests to perform depend on each case: different people find different usability problems. According to a study performed by Nielson and Landauer[57], typically 5 tests users are enough to identify major problems with a system, finding around 75 percent of the usability problems with the interface. Systems were the usability is a critical factor, the test effort needs to be superior in order to exhaustively test the system and guarantee it maximum performance and efficiency.

The participants should be selected in order to constitute, as much as possible, a representative sample of users.

9.2 Experimental Aspects

This section covers the usability test planning. First, the experimental plan is presented. Then, the profile of the participants in the experience is disclosed. Afterwards we explain the methodology used to perform this test.

The Test Plan can be found on Appendix C.

9.2.1 Experimental Plan

The main goals of this usability test is to analyse the efficiency of use and the subjective satisfaction of the user with the developed data visualizations for entity Linking, in order to identify the most promising data visualization and how it can be improved. This procedure evaluates the product by testing it on users.

This usability test use 3 different methodologies:

1. User tasks

2. Final Questions
3. SUS evaluation

The test will take place in a controlled environment and the material for the test (laptop, mouse and questionnaire) will be provided to the users. To get some posterior metrics, a screen recorder tool will be used to record the session.

To assess the experimental procedure, a pilot session was conducted with two volunteers were as testers. This procedure allowed the evaluator to train how to approach the experimental subjects, verify that the tasks are reasonable, verify that the description of system and tasks are understandable, improve the test flow, estimate the time needed to accomplish all the tasks and test the screen recorder tool.

All the users in the experience, participate voluntarily and gave their informed consent. User data and information collected will be used anonymously.

Before the test was applied, a brief verbal introduction to the subjacent motivation was made and it was explained what is the role of the user in the test. This was followed by an introduction to the test, disclosing its content and duration. It was stressed out that the user performance and the ability of the user interact with the system is not the focus of the test, but the system by itself.

9.2.2 User Profile

To perform usability evaluation of the Entity Linking system, a test group of nine elements was used. Two of these elements were used to perform a pilot session of the test, while the others constitute the test group.

Elements of the test group were chosen, as much as possible, in order to form a heterogeneous set representative of the system's users.

Since the final users of the Entity Linking application are fraud analysts, test users were selected among the employees from Feedzai. All of the users are male, with ages ranging 24 to 31. Since the number of available subjects to perform the test in the summer was low, it was not possible to get a more heterogeneous test group.

Three different sets of users can be distinguished, according with their experience in data analysis and fraud detection:

- Inexperient user: user with less than 6 months of experience in fraud detection and not an expert in data analysis;
- Fraud Analyst: user with more than 6 months of experience in fraud detection but not an expert in data analysis;
- Data Analyst: user with more than 6 months of experience in fraud detection and an expert in data analysis;

Three elements of the test group were classified as inexperient users, two elements as fraud analysts and two other elements as data analysts

9.2.3 Methodology

Usually, in a usability test, tasks are graded according with their importance and degree of doubt, to choose the top-ranking tasks to be implemented on the test. However, the goal of this test is not the tasks, *per si*, but the visualizations.

Since not all visualizations can be tested, a selection was made according with those parameters. Visualizations were rated by their importance and degree of doubt on their implementation, in a scale from 1 to 6, where 1 correspond to the lowest value and 6 to the highest. Classification of the visualization is obtained by multiplication of these two factors. Data visualization classification can be found on table 4.

Visualization	Importance	Degree of Freedom	RESULT
Table	1	1	**1**
Matrix	6	3	**18**
Geographical Referencing	2	3	**6**
Force Directed Graph	6	6	**36**
Circular Diagram	6	5	**30**
Chord Diagram	4	3	**12**

Table 4: Classification of data visualizations according with their importance and degree of freedom to be implemented

The highest classified data visualizations were Force Directed Graph, Circular Diagram and Matrix. Therefore, they will be the subjects of the test.

An initial task concerning the table will be used to contextualize the test. Due the public familiarity with this type of visualization, it can be used for comparison. In each visualization under analysis, the user will analyse a non-fraudulent transaction and a fraudulent transaction and compare both results.Detailed enunciation of the test tasks can be found in the Test Plan, on Appendix C.

9.2.4 Questionnaire

Brief questions will be performed after the test, in order to obtain insight about the system and the experience. The questions to be made are:

1. In your opinion, which data visualization was your favourite and why?
2. What could make it better?
3. Point out 2 positives and 2 negatives aspects of this experience.
4. Do you want to add anything or make any question?

9.2.5 System Usability Scale

To measure the usability of the system, the System Usability Scale [60] will be used. This questionnaire consists of a 10 item closed answer questions, with five response

options that range from “strongly agree” to “strongly disagree”.

To avoid the natural tendency to provide a neutral answer, this questionnaire was modified. An additional option for respondents was included, in order for the scale to have an even number of points (figure 44).

Items on the questionnaire were selected so that half of them elicit positive answers and the other half negative answers. Positive items are alternate by negative items, in order to prevent response biases caused by respondents not having to think about each statement.

	The System Usability Scale Standard Version	Strongly disagree					Strongly agree
		1	2	3	4	5	6
1	I think that I would like to use this system.	O	O	O	O	O	O
2	I found the system unnecessarily complex.	O	O	O	O	O	O
3	I thought the system was easy to use.	O	O	O	O	O	O
4	I think that I would need the support of a technical person to be able to use this system.	O	O	O	O	O	O
5	I found the various functions in the system were well integrated.	O	O	O	O	O	O
6	I thought there was too much inconsistency in this system.	O	O	O	O	O	O
7	I would imagine that most people would learn to use this system very quickly.	O	O	O	O	O	O
8	I found the system very cumbersome to use.	O	O	O	O	O	O
9	I felt very confident using the system.	O	O	O	O	O	O
10	I needed to learn a lot of things before I could get going with this system.	O	O	O	O	O	O

Figure 44: System Usability Scale

9.3 Results

The following section presents the results obtained in the usability evaluation.

9.3.1 Task Performance

To analyse task performance, the following metrics used were: ability to interpret each data visualization correctly; time spent on each task; ability to interpret each data visualization correctly; ability to distinguish abnormal patterns from normal ones.

All users were able to correctly interpret the data displayed in each visualization. Likewise, all the users were able to correctly identify patterns out of the ordinary in all data visualizations. However, inexperients users had more difficulties to correlate the data with fraud analysis, while data scientists excelled on it.

The average amount of time spent on each task is displayed on the graphic of figure 45. Time measured covers the time spent by the user while performing the tasks and analysing the visualization.

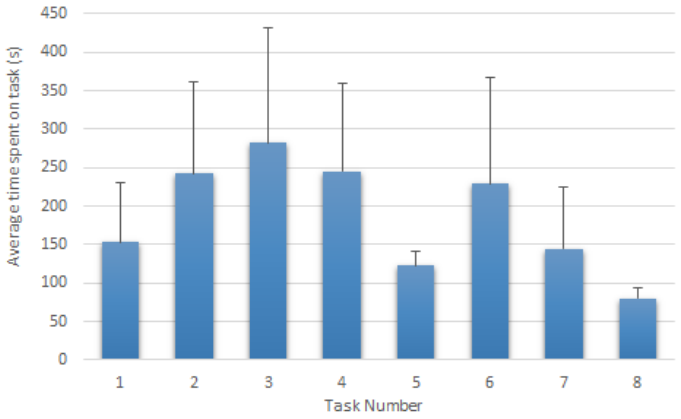

Figure 45: Time spent, on average, by the test users on each task performed

Time spent exploring table visualization (task1) was, on average, 150s. The first contact with each data visualization (task2, 3 and 4) ranged, on average from 240s to 280s, with more time being spent on the force directed graph visualization. Time spent on each visualization on the second contact decreased linearly from the force directed graph (task 6, with 229s, on average) to the circular diagram (task 7 - 145s) and to the matrix diagram (task 8 - 80s).

Although no significant difference was found on the time spent on each task by different group of users, fraud analysts tend to spend more time on each task. But, instead of reflecting the difficulty of the task, that extra time was spent exploring other options out of the scope of the test.

9.3.2 Questionnaire

The first item in the questionnaire intended to assess user's preferences regarding the data visualizations. The majority of the users (five in seven) elected the Force Directed Graph as the data visualization of choice to analyse linked entities and detect fraud. Circular diagram come as the second favourite visualization, with two users preferring it to the Force Directed Graph. The matrix visualization was the less preferable visualization.

Some of the users pointed out that, more than supersede one another, these data visualizations complement each other. Several users indicated that the circular diagram might be preferable to use when the number of transaction is small. Matrix visualization also raised favourable critics when it comes to identify clusters and to analyse single-parameter relationships.

Several suggestions were made to improve the platform and the data visualizations. Regarding the platform, one of the users suggested ordering the parameters alphabetically. Some of the users complained about the names "Select" and "Relate"

which don't fully represent their function, therefore make the platform harder to interact with. Data volume limitation on the data visualizations was also a source of complaint, with two users suggesting to condense data according with a parameter (for example, transactions from the same Customer ID), to be able to analyse a higher volume of data with less ligations.

Concerning matrix diagram, the majority of the users complained about the difficulties in understanding the data represented when more than one parameter was involved. One suggestion was to, inside a cluster, order transactions by cluster or frequency, to try to make sense of the other relationships represented in the matrix.

Interesting suggestions were made regarding force directed graph. One of the users suggested to add another filter in the menu to group transactions in a single node, in which the size the node would represent the number of transactions grouped. Another user recommended, beside the colour parameter, encode another parameter by shape. Therefore, two different dimensions, such as Customer ID and Card, could be captured in the same visualization. Other advice was to add the possibility of selecting a node and, when that happened, its edges would be highlighted. Another user advocated that the fisheye could be used to add more information to the graph: when applied over nodes or edges, additional information, such as symbols representing the parameter responsible for the relationship, would become visible. Besides that, it was suggested that the tooltip would open with a mouse click on a node, and would stay open until deliberately dismissed by the user.

To improve the circular diagram, it was suggested to remove the transactions IDs and replace them by a single bar, representing the first level hierarchy. Then, a single parameter could be encoded by the ligations, through different colours.

The following question, requested that the users pointed out 2 positive and 2 negative aspects of this experience. The main positive aspects shared by the users were:

- Data visualizations are more comprehensible than tabular view of data;
- Different data visualizations have different advantages and disadvantages, but conclusions can be easily draw;
- Abnormal patterns are easily distinguishable;
- Jumping from data visualization to data visualization is a good experience, since each data visualization allows to immediately perceive different information over the same data.

The main negatives aspects of the experience identified by the users were:

- Data visualizations take too long to process;
- Volume of data is low;
- Hard to understand without help;
- Some data visualizations should be improved to become more efficient;
- Too much information in a data visualization saturate the user.

9.3.3 System Usability Scale Evaluation

The System Usability Scale (SUS) provides a "quick and dirty", reliable tool for measuring the usability of a system. Each item of the test was scored from 0-40 and then multiplied by 2.5 to convert it to a scale from 0-100. However, results from SUS should not be interpreted as percentages. Instead SUS results can be classified according with figure 46.

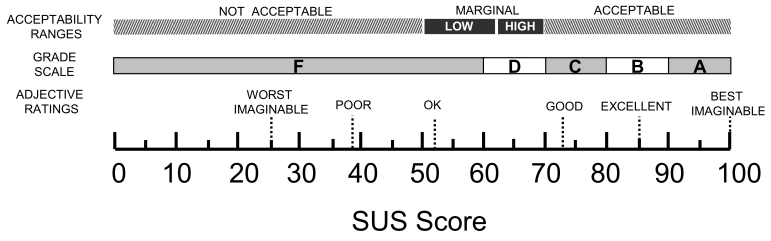

Figure 46: Comparison of the adjective ratings, acceptability scores, and American school grading scales, in relation to the average SUS score [13].

The result from this analysis was positive, obtaining a final SUS score of 68,9 points. This score represents a D grade with an adjective rating of Good.

9.4 Suggestions and Recommendations

Based on the feedback from the users, to improve the usability of the system and data visualizations effectiveness, the following modifications to the system can be suggested:

- System Parameters: When parameters are listed, order them alphabetically;
- System Options: Change the name of options "Select" and "Relate" to a more significant name;
- Matrix: inside a cluster, order transactions by cluster or frequency;
- Force directed graph:
  - Encode two-parameter relationships by colour and shape;
  - Highlight edges of selected node
  - Additional information concerning nodes and edges be displayed over fisheye magnification eye
  - Tooltip on click
- Circular Diagram: represent first level hierarchy parameters by a group bar and use colourful links to represent single-parameter relationships.

10 Future Work

The intent of this chapter is to consolidate the work done by exploring potential directions for this project. Focus will be on the system and in the three most promising data visualizations for Entity Linking (matrix, force directed graph, circular diagram) and how they could be improved, considering the feedback obtained on the usability tests.

10.1 Graph Database

One of the main problems found on the Entity Linking application was the time spent processing relationships between transactions. The use of a graph database could overcome this problem. A graph database is a sort of database that stores information as a graph, with nodes, edges and properties. Therefore, relationships are stored as data, allowing nodes to be linked together easily in a single operation. Complex hierarchical structures can easily be accessed and retrieved. Regardless of the dataset size, graph databases excel in managing highly connected data, performing complex queries efficiently.

10.2 Matrix

Matrix diagram could be improved by creating clusters inside the clusters. That could be achieved by adding a second clustering parameter. The resulting visualization would be cleaner and easier to interpret (figure 47).

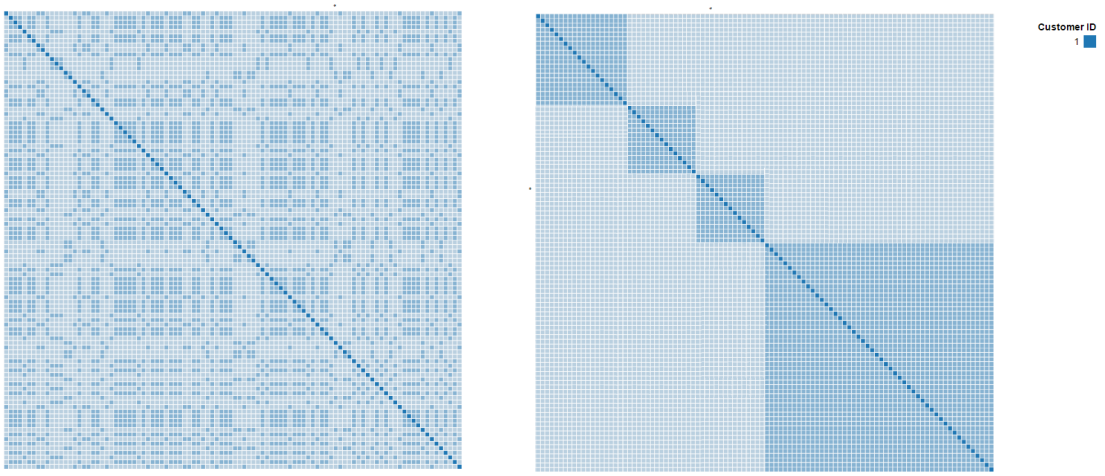

Figure 47: Comparison between the actual Matrix (left) and a Matrix with two levels of clustering.

Data comprise transactions from a single client using four different credit cards.

10.3 Force Directed Graph

To improve Force Directed Graph visualization, different shapes could be used to represent a second parameter (figure 48). Therefore, two different parameters can be easily been distinguishable on the graph, providing more information in a single analysis.

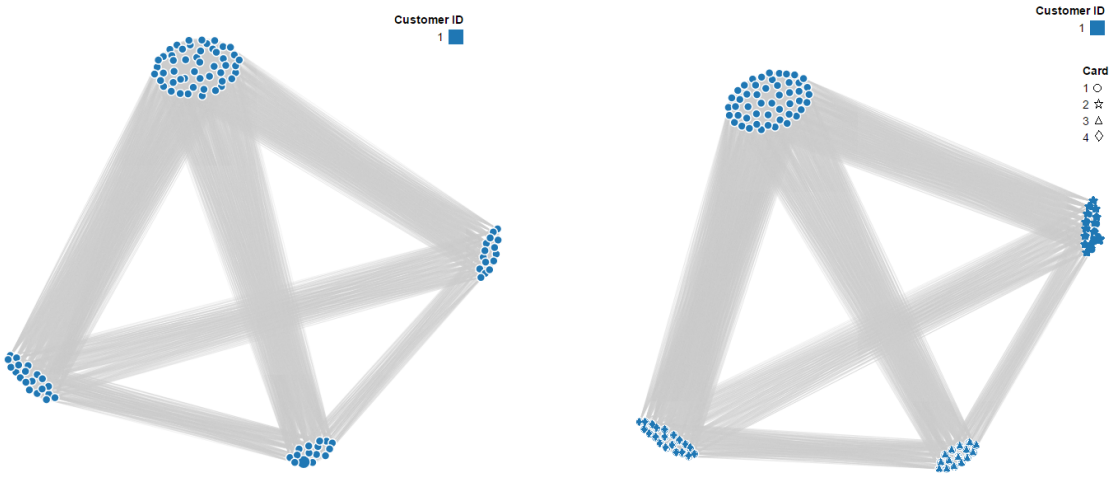

Figure 48: On the left, force directed graph with colour encoding and, on the right, force directed graph with colour and shape encoding (right).
Data comprise transactions from a single client using four different credit cards.

10.4 Circular Diagram

One of the main problems of the circular diagram is its readability. This diagram provides a lot of visual information and, part of it, it's encoded on the labels with the transaction ID displayed for each transaction. This renders the visualization cumbersome.

To improve its aspect, the labels containing the transaction ID could be replaced by a label indicating the value of the hierarchical parameter.

Figure 49 proposes a model where labels for the two hierarchical parameters, used to distribute the transactions on the diagram, are disposed around the circular diagram, in a ring-shaped fashion.

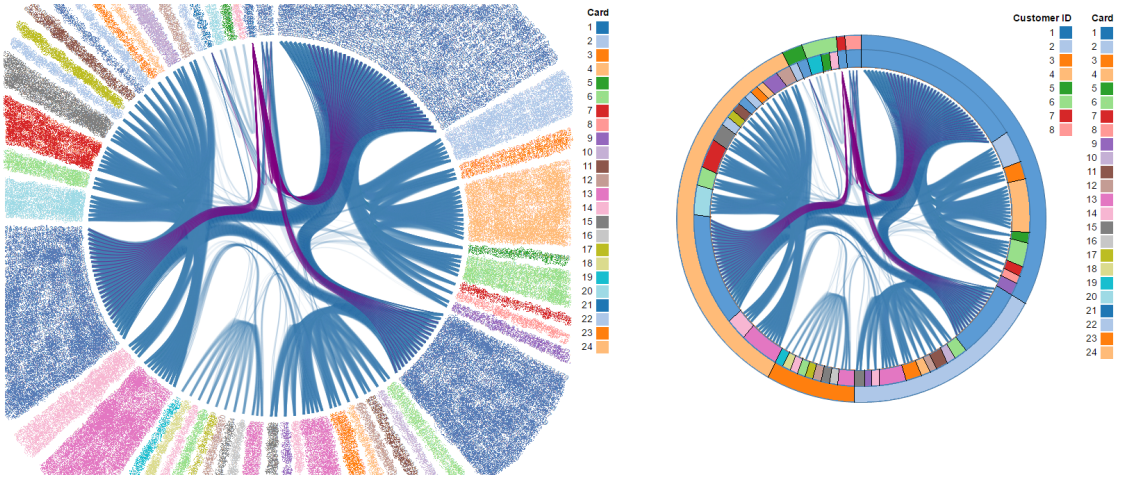

Figure 49: Circular diagram discriminating each transaction is represented on the left (labels were scrapped for confidentiality reasons). On the right, a configuration labelling the hierarchical parameters in a ring-shaped version around the diagram. Included transactions represent several customers sharing multiple credit cards.

11 Conclusion

This dissertation aims to address the problem of Entity Linking in fraud analysis. However, the task of Entity Linking is not trivial and the huge amount of data available to analyse adds even more complexity to this task.

It is easier to make sense of extensive amounts of data when presented graphically. Data visualizations can help comprehend the information and identify relationships and patterns. Therefore, they are the focus of this project. However, the purpose of this dissertation was not the build of a data visualization *per si*, but of an exploratory tool to help the designers of a fraud detection system to extend their capabilities through data visualizations. This tool introduces a set of different visualizations for Entity Link analysis and offers several options to refine them and tune them, according with the needs of the fraud analysts.

This endeavour proved to be quite challenging. It required a deep research investment to acquire background knowledge in electronic fraud detection and prevention and in data visualizations, as well as an understanding the state of the art. To develop the system, it was necessary to became acquainted with a set of new technologies and frameworks. Moreover, there was an ongoing care with the engineering process behind the development of the software system.

Different data visualizations were developed and refined for Entity Linking, namely: table, matrix diagram, geographical referencing, force directed graph, circular diagram and chord diagram. From those, the matrix, force directed graph and the circular diagram were disclosed as the visualization with most potential to be used for fraud detection.

Tests with users further accessed their efficiency as fraud detection tools. Consensual opinion was that abnormal patterns were easily distinguishable in every data visualization. What differentiate them, was not their capability as fraud detection tools, but their ability to convoy information.

Furthermore, these tests brought to light several improvements that could be made in each visualization. Some of those suggestions were prototyped and introduced on this report, but the field of data visualization is too proliferous, and there are several of interesting paths yet to be explored.

Although the force directed graph was clearly the public favourite, user's opinion opened up the discussion for combining different data visualizations for Entity Linking, since the insights provided by each visualization are different and complement each other.

Therefore, the hunt for the perfect Entity Linking data visualization to be integrated with Feedzai's Alert Manager continues. However, the exploratory tool built in this dissertation was a success and it will be fundamental for that process.

References

[1] Bootstrap. http://getbootstrap.com/.

[2] Data-driven documents. https://d3js.org/.

[3] jquery. https://jquery.com/.

[4] Natural earth. http://www.naturalearthdata.com/.

[5] Detecting and preventing fraud with data analytics. Technical report, 2013.

[6] The role of data analytics in fraud prevention. Technical report, February 2014.

[7] Elanders Americas. Why is ecommerce so important? https://www.elandersamericas.com/Pages/why_ecommerce.aspx.

[8] Business Analysis. Moscow method for requirements prioritization. http://www.businessanalysis.in/2013/06/moscow-method-for-requirements.html.

[9] Bair Analytics. Introduction to link analysis. http://www.bairanalytics.com/community/blog/introduction-to-link-analysis-part-1-time-event-charts/, 2014.

[10] Bob Angus. How to choose a fraud prevention service, for ecommerce. *Practical eCommerce*, December 2014. "http://www.practicalecommerce.com/articles/76462-How-to-Choose-a-Fraud-Prevention-Service-for-Ecommerce".

[11] Atlassian. Confluence. https://www.atlassian.com/software/confluence.

[12] Atlassian. Jira software. https://www.atlassian.com/software/jira.

[13] Aaron Bangor, Philip Kortum, and James Miller. Determining what individual sus scores mean: Adding an adjective rating scale. *Journal of Usability Studies*, May 2009.

[14] Dan Barta and David Stewart. A layered approach to fraud detection and prevention. Technical report.

[15] Antonia Bertolino. Software testing research: Achievements, challenges, dreams. *Future of Software Engineering*, July 2007.

[16] Tony Bhe, Peter Glasmacher, Jacqueline Meckwood, Guilherme Pereira, and Michael Wallace. *Event Management and Best Practices*. IBM, 2004.

[17] Bigcommerce. Bigcommerce: Ecommerce software & shopping cart platform. https://www.bigcommerce.com/.

[18] Brian Bimschleger. How to tell a powerful story with data visualization. https://www.thinkwithgoogle.com/articles/tell-meaningful-stories-with-data.html, October 2014.

[19] Mike Bostock. Topojson. https://github.com/mbostock/topojson.

[20] William Brinton. *Graphic Presentation*. PreLinger Library, 1939.

[21] The Free Dictionary by Farlex. Fraud definition. http://legal-dictionary.thefreedictionary.com/fraud.

[22] Adrian Bănărescu. Detecting and preventing fraud with data analytics. *Procedia Economics and Finance*, 32:1827 – 1836, 2015. Emerging Markets Queries in Finance and Business 2014, {EMQFB} 2014, 24-25 October 2014, Bucharest, Romania.

[23] P. Christen and R. Gayler. Towards scalable real-time entity resolution using a similarity-aware inverted index approach. In *Proceedings eventh Australasian Data Mining Conference (AusDM 2008), Glenelg, South Australia*, pages 51–60, 2008.

[24] Peter Christen. *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Springer Publishing Company, Incorporated, 2012.

[25] Dr. Stephen Coggeshall. I see fraud rings. Technical report, November 2012.

[26] Mike Cohn. *User Stories Applied for Agile Software Development*. Addison-Wesley Professional, 2004.

[27] Alex Cowan. Your best agile user story. http://www.alexandercowan.com/best-agile-user-story/, February 2014.

[28] Pete Deemer, Gabrielle Benefield, Craig Larman, and Bas Vodde. A lightweight guide to the theory and practice of scrum. 2012.

[29] Paul Demery. Online fraud costs e-retailers $3.5 billion in 2012. https://www.internetretailer.com/2013/03/28/online-fraud-costs-e-retailers-35-billion-2012.

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

[31] EMC. E-commerce fraud trends 2014: Securing the online shopping cart. June 2014.

[32] Tom Fawcett. Adaptive fraud detection. July 1997.

[33] Feedzai. Feedzai: Fraud prevention powered by machine learning. http://feedzai.com/.

[34] Feedzai. Fraud prevention api. http://dev.feedzai.com/rest-api/.

[35] Legal Information Institute from Cornell University Law School. Computer and internet fraud: An overview. https://www.law.cornell.edu/wex/computer_and_internet_fraud.

[36] Adam Graycar and Russell Smith. Identifying and responding to electronic fraud risks. *30th Australasian Registrars' Conference*, November 2002.

[37] Robert Grossman. Alert management systems: A quick introduction. July 2003.

[38] Yuhang Guo, Wanxiang Che, Ting Liu, and Sheng Li. A graph-based method for entity linking. In *In Proc. IJCNLP2011*, 2011.

[39] John Hayes. The importance of outliers. `https://savionline.wordpress.com/2013/07/24/the-importance-of-outliers/`.

[40] Monika Henzinger. Link analysis in web information retrieval. *IEEE Data Engineering Bulletin*, 23:3–8, 2000.

[41] Wei Hu, Honglei Qiu, and Michel Dumontier. *The Semantic Web - ISWC 2015: 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part II*, chapter Link Analysis of Life Science Linked Data, pages 446–462. Springer International Publishing, Cham, 2015.

[42] Yifan Hu. Efficient, high-quality force-directed graph drawing. *The Mathematica Journal*, October 2006.

[43] Mathieu Jacomy, Tommaso Venturini, Sebastien Heymann, and Mathieu Bastian. Forceatlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software. *PLOS one*, June 2014.

[44] Kevin Jiang. Online retailers: How to recognize and reduce ecommerce fraud. `https://www.trulioo.com/blog/2015/08/05/online-retailers-how-to-recognize-and-reduce-ecommerce-fraud`, August 2015.

[45] Jeff Sutherland Ken Schwaber. The scrum guide. July 2013.

[46] Philippe Kruchten. The 4+1 view model of architecture. *IEEE Softw.*, 12(6):42–50, November 1995.

[47] Jeremy Kubica, Andrew W. Moore, David Cohn, and Jeff G. Schneider. Finding underlying connections: A fast graph-based method for link analysis and collaboration queries. In *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA*, pages 392–399, 2003.

[48] Ash Kyd. Download vector maps. `https://geojson-maps.kyd.com.au/`.

[49] Business Analyst Learnings. Moscow : Requirements prioritization technique. `http://businessanalystlearnings.com/ba-techniques/2013/3/5/moscow-technique-requirements-prioritization`.

[50] LexisNexis. Merchants contend with increasing fraud losses as remote channels prove especially challenging. September 2015.

[51] Francois-Serge Lhabitant. Correlation vs. trends: A common misinterpretation. *EDHEC-Risk Institute*, April 2011.

[52] PKF Littlejohn. The financial cost of fraud. http://www.pkf-littlejohn.com/the-financial-cost-of-fraud-2015.php. Accessed: 2016-05-10.

[53] Magento. Fraud and chargeback detection. prevent fraud. https://www.magentocommerce.com/magento-connect/fraud-and-chargeback-detection-prevent-fraud.html.

[54] Isabel Meirelles. *Design for Information: An Introduction to the Histories, Theories, and Best Practices Behind Effective Information Visualizations*. Rockport Publishers, october 2013.

[55] Tamara Munzner. *Visualization Analysis and Design*. A.K. Peters visualization series. A K Peters, 2014.

[56] Lan Nie, Brian D. Davison, and Xiaoguang Qi. Topical link analysis for web search. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 91–98, New York, NY, USA, 2006. ACM.

[57] Jakob Nielson. How to conduct a heuristic evaluation. https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/, January 1995.

[58] Humphrey Waita Njogu, Luo Jiawei, Jane Nduta Kiere, and Damien Hanyurwimfura. A comprehensive vulnerability based alert management approach for large networks. April 2012.

[59] Service Now. Incident alert management. http://wiki.servicenow.com/index.php?title=Incident_Alert_Management.

[60] U.S. Department of Health & Human Services. System usability scale (sus). https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html.

[61] U.S. Department of Health & Human Services. Usability - improving the user experience. http://www.usability.gov/.

[62] Australian Bureau of Statistics. Statistical language - what are variables? http://www.abs.gov.au/websitedbs/a3121120.nsf/home/statistical+language+-+what+are+variables.

[63] Jeff Paine. Ngeojson-topojson. http://jeffpaine.github.io/geojson-topojson/.

[64] Andrei Pandre. What is the outlier? https://apandre.wordpress.com/visible-data/outliers/.

[65] Subrata Paul. On some aspects of link analysis and informal network in social network platform. 2, 2013.

[66] PayMill. 3 ways to help prevent fraud on your e-commerce website. https://blog.paymill.com/help-prevent-fraud/, March 2015.

[67] Scotland Police. Fraud. https://http://www.scotland.police.uk/keep-safe/advice-for-victims-of-crime/fraud.

[68] Mike Potel. Mvp: Model-view-presenter the taligent programming model for c++ and java", taligent inc. Technical report, 1996.

[69] The Fraud Practice. Common fraud schemes. http://www.fraudpractice.com/fl-fraudscheme.html.

[70] Aaron Press. The three merchant categories most vulnerable to web-related fraud. https://www.internetretailer.com/commentary/2015/12/18/three-merchant-categories-most-vulnerable-fraud, December 2015.

[71] FraudLabs Pro. Fraudlabs pro - fraud detection & fraud prevention solutions. http://www.fraudlabspro.com/.

[72] Merchant Protector. Merchant protector - fight fraudulent orders. https://www.merchantprotector.net/.

[73] Delip Rao, Paul McNamee, and Mark Dredze. *Multi-source, Multilingual Information Extraction and Summarization*, chapter Entity Linking: Finding Extracted Entities in a Knowledge Base, pages 93–115. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[74] Riskified. Riskified: ecommerce fraud & chargeback prevention solution. http://www.riskified.com/.

[75] Gorka Sadowksi and Philip Rathle. Fraud detection: Discovering connections using graph databases. Technical report, January 2015.

[76] Zakaria Saleh. The impact of identity theft on perceived security and trusting e-commerce. *Journal of Internet Banking and Commerce*, August 2013.

[77] Nagiza F. Samatova, William Hendrix, John Jenkins, Kanchana Padmanabhan, and Arpan Chakraborty. *Practical Graph Mining with R.* Chapman & Hall/CRC, 2013.

[78] SAP. Sap alert management overview. http://scn.sap.com/docs/DOC-14199.

[79] SAS. Data visualization: What it is and why it's important. http://www.sas.com/en_sg/insights/big-data/data-visualization.html.

[80] Sift Science. Sift science: Machine learning fraud detection & prevention. https://siftscience.com/.

[81] SecureBuy. Securebuy - magento 2013 fraud report. Technical report, 2013.

[82] Shopify. Shopify. https://www.shopify.com/.

[83] Signifyd. Shopify plus video 4. https://www.youtube.com/watch?v=icGtBT-RqjQ.

[84] Signifyd. Signifyd - fraud protection & chargeback prevention for ecommerce. https://www.signifyd.com/.

[85] Crime Tech Solutions. Link analysis and fraud. https://fightfinancialcrimes.com/2015/12/17/link-analysis-and-fraud/, December 2015.

[86] Josh Sorbel. Identity theft and e-commerce web security: A primer for small to medium sized businesses. *GIAC Security Essentials Certification*, November 2003.

[87] Subuno. Subuno. http://www.subuno.com/how-it-works/.

[88] John R. Talburt, Yinle Zhou, and Savitha Yalanadu Shivaiah. SOG: A synthetic occupancy generator to support entity resolution instruction and research. In *Proceedings of the 14th International Conference on Information Quality, ICIQ 2009, Hasso Plattner Institute, University of Potsdam, Germany, November 7-8 2009*, pages 91–105, 2009.

[89] CA Technologies. What is alert management system (ams)? https://supportconnectw.ca.com/public/impcd/r11/Administration_and_Maintenance/doc/What%20is%20Alert%20Management%20System.pdf.

[90] Stat Trek. How to describe data patterns in statistics. http://stattrek.com/statistics/charts/data-patterns.aspx.

[91] T.B.T. Truong, F. Frizon de Lamotte, J-Ph. Diguet, and F. Said-Hocine. Alert management for home healthcare based on home automation analysis. September 2010.

[92] UniBul. How to minimize fraudulent e-commerce transactions. http://blog.unibulmerchantservices.com/how-to-minimize-fraudulent-e-commerce-transactions/.

[93] Daniel Waisberg. Tell a meaningful story with data. https://www.thinkwithgoogle.com/articles/tell-meaningful-stories-with-data.html, March 2014.

[94] VM Ware. Alerts and alert definitions. https://pubs.vmware.com/vfabric5/index.jsp?topic=/com.vmware.vfabric.hyperic.4.6/Alerts_and_Alert_Definitions.html.

[95] Christopher Westphal. *Data Mining for Intelligence, Fraud and Criminal Detection: Advanced Analytics and Information Sharing Technologies*. CRC Press, 2008.

[96] WorldPay. Fragmentation of fraud. November 2014.

A Alert Manager User Stories

In this project, we chose to capture the Alert Management requirements using user stories. This process was made in collaboration with the stakeholders of the project and took in consideration the analysis of other applications with Alert Management functionalities in Fraud Detection. In this appendix we'll disclose the methodology used, the entities considered and the user stories identified.

A.1 Methodology

This subsection describes the template used to describe a user story, the format of associated test cases and the prioritizing method used.

A.1.1 User Stories

A user story[26] is an agile software development tool used to describe a software feature from an end-user perspective, creating a simplified description of a requirement. Besides providing a simple way to formalize the requirements, user stories stimulate design thinking. Moreover, their small dimension allows modifications to be make quickly, in case the requirements are altered.

The template for a user story uses the following format:

As a ***<role>****, I want* ***<feature>*** *so that* ***<benefit>****.*

A.1.2 Test Cases

To complement, test cases[27] can be associated to user stories. These tests validate the output of the development process of the corresponding user story, guaranteeing that the story delivers what is intended. The format used for test cases is:

Make sure that ***<condition>****.*

A.1.3 MoSCoW Method

Prioritizing the User Stories helps to determine which stories are more important to implement and which are the least. The MoSCoW method [8, 49] defines four different priority levels:

- Must (M): requirement that must be satisfied in the product in order for it be accepted;
- Should (S): requirement with high-priority that should be included in the final solution if possible, according with the available time-frame;

- Could (C): desirable requirement that could be nice-to-have; however, the application is still accepted without featuring this functionality;
- Won't (W): desirable requirement that will not be implemented in the current version of this project.

A.2 Entities and roles

Three different types of entities were identified, in Alert Management application, which perform different roles:

- Data Analyst – The role of the data analyst is to explore the data and build the dashboard that better conveys its story.
- Developer – The role of the developer is to author new widgets and to integrate the embedded dashboards into custom web pages.
- Fraud analysts – The role of the business user is to manage risk internally in a company, using the information presented in the dashboards to make informed decisions.

User stories were analyzed in the point of view of the fraud analyst, which is the final user of the application.

A.3 User Stories

The user stories for the Alert Manager application were grouped in modules:

I Overview;

II Entity Profile;

III Entity Linking;

IV Geographical Information;

V Geographical Profiling.

Henceforth, user stories and respective case tests will be presented for each module.

US01	Display Fraud Detection Statistics
Priority	Won't
Context	When I am in the Alert Manager dashboard.
Story	As a fraud analyst, I want to know the number of fraudulent orders detected by the fraud detection system at my company, compared to the total orders made, and the number of orders manually marked as fraudulent, compared to the numbers of alerts generated, in a period of time, so I can have an overview of the system's operation.
Test	Make sure it is possible to select a period of time. Make sure it is possible to display the number of fraudulent and non-fraudulent orders on that period of time. Make sure it is possible to display the number of fraudulent, non-fraudulent orders and alerts generated in that period of time. Make sure it is possible to display the number of alerts reviewed manually and marked as fraudulent and non-fraudulent, on that period of time.
US02	Analyze all the Information related with an order
Priority	Won't
Context	When I am in the Alert Manager dashboard.
Story	As a fraud analyst, I want to check all Customer, Transaction, Merchant, Payment and Device information, so I can access all the information regarding a specific order.
Test	Make sure it is possible to display all Customer information. Make sure it is possible to display all Transaction information. Make sure it is possible to display all Merchant information. Make sure it is possible to display all Payment information. Make sure it is possible to display all Device information.

Table 5: User stories for the Alert Manager application in the Overview module.

US03	Examine customer activity
Priority	Must
Context	When I am analyzing a specific alert.
Story	As a fraud analyst I want to be able to visualize the user activity in the system (when the account was created and accessed, when the user performed transactions, when was the last login, how long does it take for this user to do a transaction after he has entered the system and the maximum time of his absence on the system), in order to understand if this is an old customer or a new one and if this is regular costumer or a sporadic one.
Test	Make sure that the information when the account was created is displayed. Make sure that the information when the account was accessed is displayed. Make sure that the information when transactions were made is displayed. Make sure that information about when the last login was is displayed. Make sure that the maximum time of this user's absence is displayed.
US04	Compare transaction data with other transactions from the customer
Priority	Must
Context	When I am analyzing a specific alert.
Story	As a fraud analyst I want to compare transaction data with other transactions from the same user in order to detect discrepancies.
Test	Make sure to display data about the transaction. Make sure to display static data from the user's history.

Table 6: User stories for the Alert Manager application in the Entity Profile module.

US05	Examine Customer information
Priority	Must
Context	When I am analyzing a specific alert.
Story	As a fraud analyst I want to check if the user uses the same personal information in all transactions (address, phone number, email, birth date), how the user's billing and shipping addresses changed over time and which devices are used by the user over time, in order to detect multiple identities for the same entity and to analyze the user's consistency.
Test	Make sure that information about the existence of multiple identities is displayed. Make sure that a list of the user's billing address, with timestamp, is presented. Make sure that a list of the user's shipping address, with timestamp, is presented. Make sure that a list of devices used by the user, with timestamp, is presented. Make sure that multiple identities for the same entity are listed.
US06	Examine merchant transactions history
Priority	Must
Context	When I am analyzing a specific alert.
Story	As a fraud analyst I want to examine all the transactions made in the store and filter by characteristics such as amounts, volume, velocity, state (accepted or rejected) in order to examine the commerce history and to identify patterns and outliers in fraud occurrences.
Test	Make sure that information of all transaction from the merchant over time is displayed. Make sure it is possible to filter that information by amount. Make sure it is possible to filter that information by volume. Make sure it is possible to filter that information by velocity. Make sure it is possible to filter that information by state.

Table 7: User stories for the Alert Manager application in the Entity Profile module.

US07	Examine user's information
Priority	Should
Context	When I am analyzing a specific alert.
Story	As a fraud analyst I want to verify if the user information (email, address, phone) is unique, or other users use it, in order to detect linked entities.
Test	Make sure that the number of users who share the same information (for each type) is presented.
US08	Check number of users per address
Priority	Should
Context	When I am analyzing a specific alert.
Story	As a fraud analyst I want to check if there are (and how many) multiple users per IP, billing or shipping address, in order to detect linked entities.
Test	Make sure that the number of users who share the same information (for each type) is presented.
US09	Check number of users per device
Priority	Should
Context	When I am analyzing a specific alert.
Story	As a fraud analyst I want to check if there are (and how many) multiple users per device, in order to detect linked entities.
Test	Make sure that the number of users who share the same information (for each type) is presented.

Table 8: User stories for the Alert Manager application in the Entity Linking module.

US10	Verify locations
Priority	Could
Context	When I am analyzing a specific alert.
Story	As a fraud analyst I want to verify how the user location (obtained from his IP), shipping address and billing address are distributed in a map in order to detect discrepancies.
Test	Make sure to display the locations on map.

Table 9: User stories for the Alert Manager application in the Geographical Information module.

US11	Verify IP/billing country match
Priority	Could
Context	When I am analyzing a specific alert.
Story	As a fraud analyst I want to verify if the IP of the user and the billing address match, in order to detect discrepancies.
Test	Make sure to verify the IP location. Make sure to verify the billing country. Make sure to present if these two locations are located in the same country.
US12	Check IP/shipping distance
Priority	Could
Context	When I am analyzing a specific alert.
Story	As a fraud analyst I want to check the distance between the IP address of the user and the shipping address, in order to detect discrepancies.
Test	Make sure to verify the IP location. Make sure to verify the shipping location. Make sure to present the distance between these two locations.

Table 10: User stories for the Alert Manager application in the Geographical Profiling module.

B User Guide

. The web application Entity Linking functions as an exploratory tool for Data Visualizations in fraud detection. On this guide, first a brief introduction to the application will be made, then it will be demonstrated how the different visualizations can be used to detect fraud.

B.1 Introducing the platform

On the main page (figure 50), a selection box allows the user to select one of the available datasets. The main page also offers an explanation on how to use the application, by detailing what each selection option is used for.

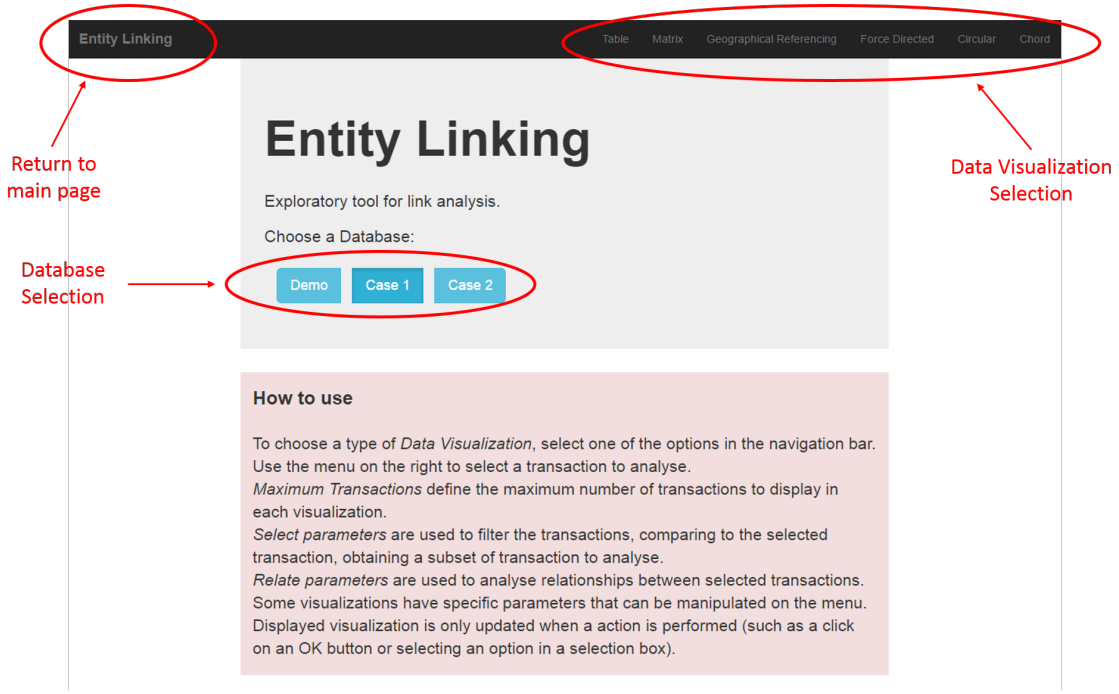

Figure 50: Entity Linking – Exploratory tool main page.

A navbar is available in every page of the application, offering the option to go back to the main page or to select a data visualization. Available data visualizations include the following graphical representations: table, co-occurrence matrix, geographical referencing, force directed graph, circular diagram and chord diagram.

The page where each data visualization is displayed follows a similar format, containing: the navbar; a menu on the right, comprising several options with which the user can interact; a main panel, where the visualization is drawn. This structure is represented in Figure 51, featuring the data visualization Table.

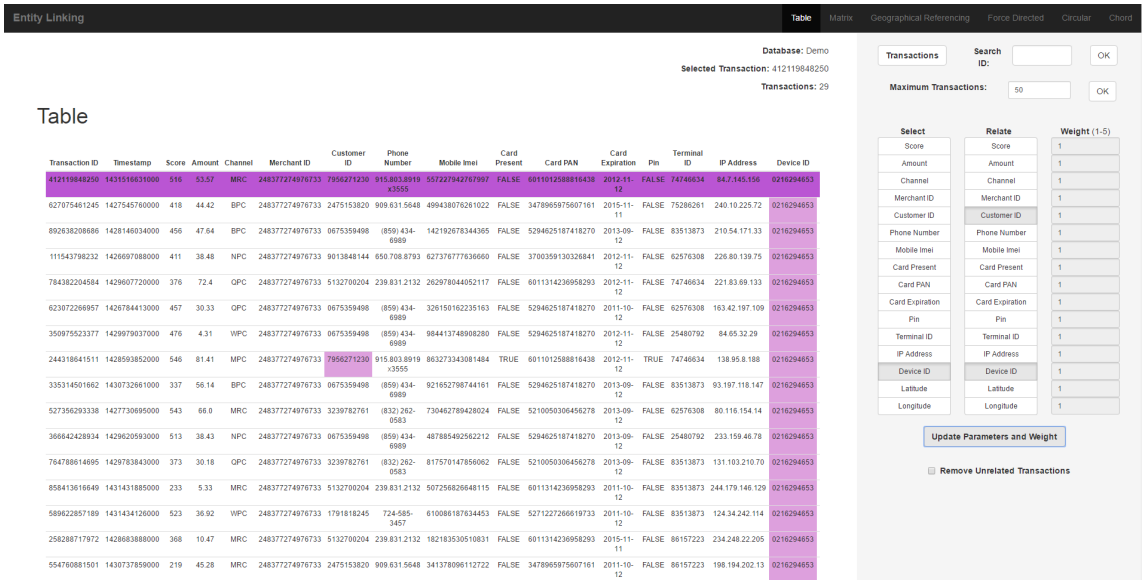

Figure 51: Example of Table data visualization.

Right Menu (figure 52) offers a set of options that are common to all data visualizations. Those are:

- Transactions: opens a pop-box with a list of IDs to select one;
- Search ID: search box, to search for a Transaction by ID;
- Maximum Transactions: input box, to manipulate the maximum number of transactions to display;
- Select: selection box, to input parameters to select transactions related with the selected one;
- Relate: selection box, to input parameters to identify relationships between the selected transactions, and to further select related transactions in the databases;
- Weight: in some visualizations (co-occurrence matrix, force directed graph and circular diagram), weight of each relation parameter can be defined as a value between 1 or 5, to make that parameter more relevant in the displayed relationship;
- Remove Unrelated Transactions: to get a cleaner visualization, checking this option removes from the visualization all transactions not directly related with the selected one.

Options specific to each data visualization are available on the right menu, in each corresponding page. Those options are displayed below the options common to all data visualization. Those specific options are:

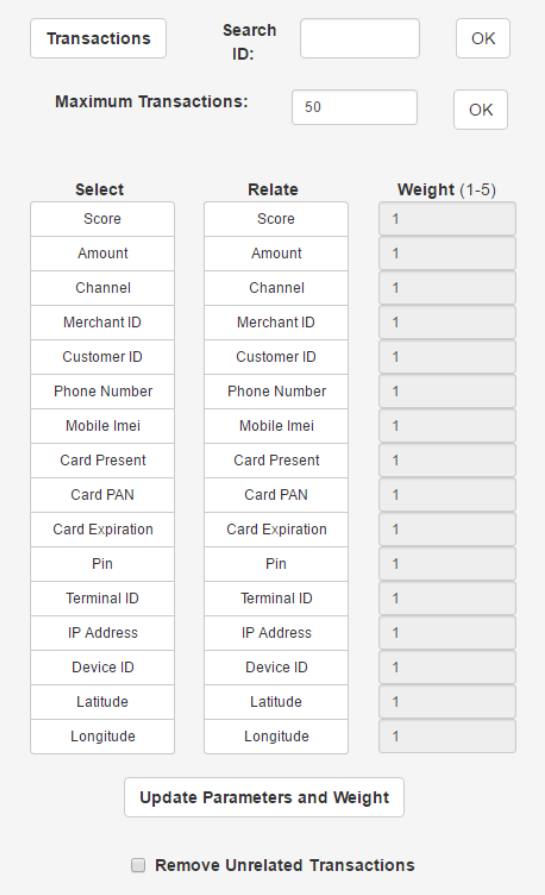

Figure 52: Right menu, available on the visualizations view, displaying options common to all data visualizations, for the user to interact with.

- Table: none;
- Matrix: grouping, ordering;
- Geographical Referencing: geographical zone;
- Force Directed Graph: Fisheye option, nodes colouring, charge, distance;
- Circular Graph: nodes colouring, hierarchy (two levels);
- Chord Diagram: grouping.

The visualization panel is where the data visualization will be displayed. On the top right corner of the panel, information regarding the visualization is presented, such as: selected dataset, selected transaction ID and number of transactions featured in the visualization. The rest of the panel is used to draw the visualization. All the available data visualizations are represented in figure 53.

In each visualization, selected transaction is emphasized, to be distinguished from the others. In the table visualization, the selected visualization row is on the top, marked with a dark purple colour, while cells featuring related parameters with it are marked as light purple. On the co-occurrence matrix, the selected transaction

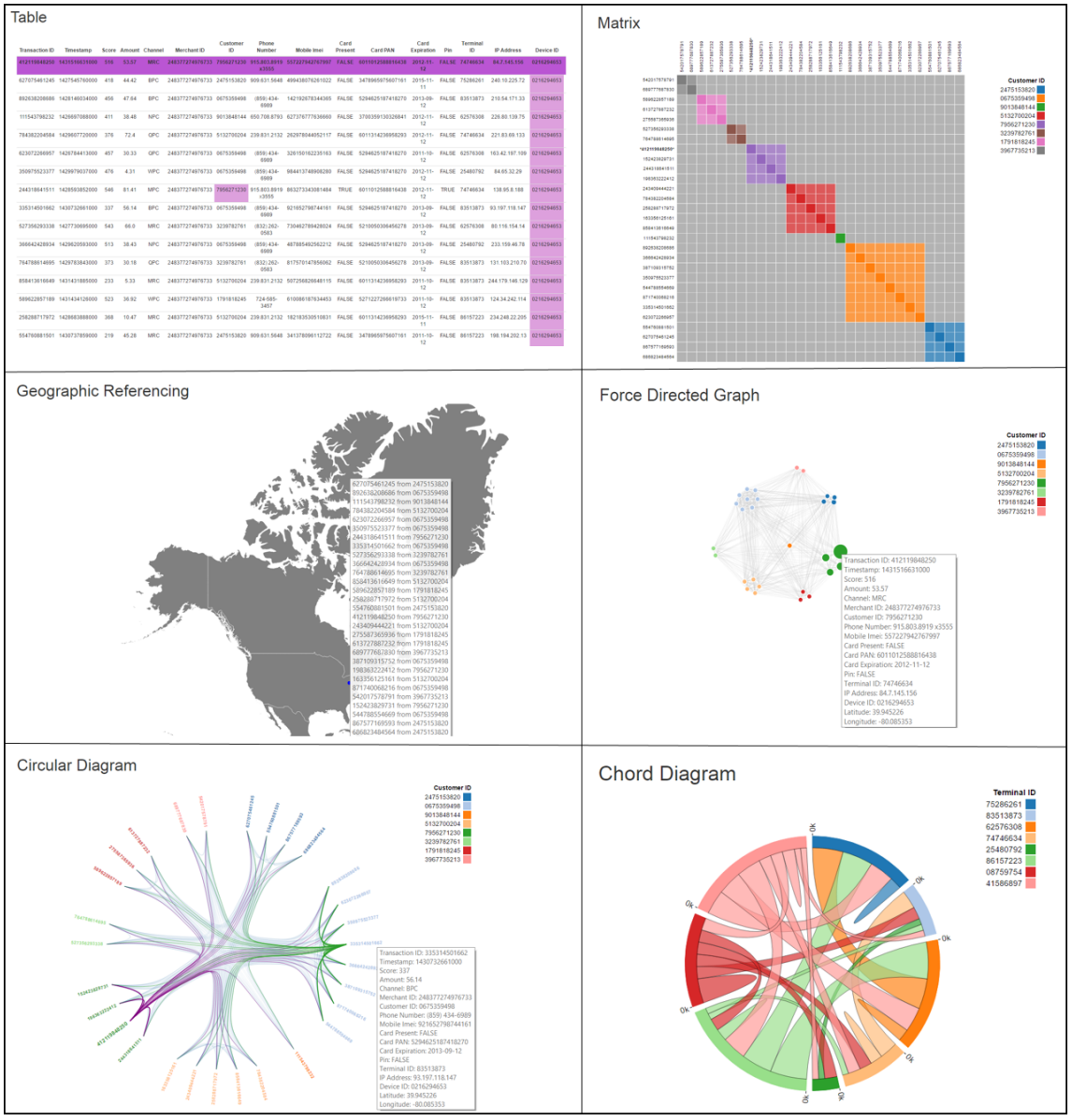

Figure 53: Representation of the available data visualizations: table, co-occurrence matrix, geographical referencing, force directed graph, circular diagram and chord diagram.

is distinguished from the others by the bold ID and the star symbol next to it. On the geographical referencing and the force directed graph, the corresponding node is bigger than the other. Circular diagram visualization distinguishes the selected transaction by its name, which is bigger and bolder than the others, and by marking purple the relations the selected transaction establishes with the others.

Code encoded visualizations display a legend next to it, indicating the parameter encoded and the values presented on the graphical representation.

All visualizations, except the table, display information regarding its elements through tooltips, as demonstrated in figure 53.

B.2 Fraud Detection

Considering a scenario where two different transactions, T1 and T2, generated alerts. To further analyse them, all transactions made with the same credit card than that transactions will be selected and relationships between transactions made by a customer with the same identification (Customer ID) or using the same credit card from those transactions will be analysed.

B.2.1 Geographical Referencing

Geographical referencing 54 shows that all translations related with T1, on those conditions, were made from Portugal, mostly around Lisbon and Porto. Transactions related with T2 also originated from Portugal, but are also related with transactions originated from Cuba. The tooltip give us more information about the transactions made in each location, but there is nothing to be concluded from this visualization.

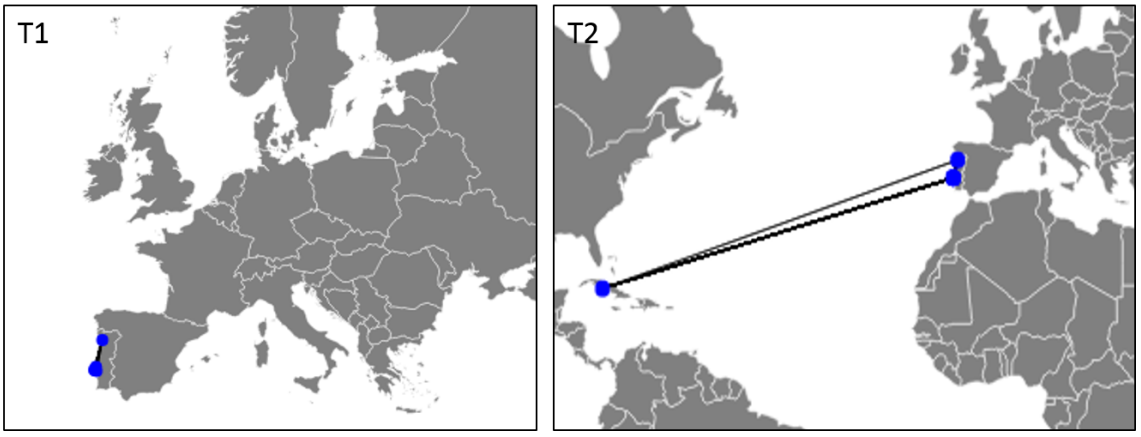

Figure 54: Geographical referencing visualization of transactions T1 (left) and T2 (right).

B.2.2 Matrix Diagram

To analyse those transactions on a matrix visualization (figure 55), it's beneficial to compare the group option by Customer ID and Card.

On transaction T1 , we can deduce that all transactions were originated by the same Customer ID and that customer used 5 different cards to perform those transactions. The majority of the transactions was made with card number 1 (represented in dark blue on figure 55, T1 b)).

Transaction T2 has a complete different pattern. There are several customers involved with selected transactions and using several cards (8 different customers IDs were identified, and 24 different credit cards). The grey pattern in the matrix indicates that all customers are using more than one card and that each cards is being used from more than one customer.

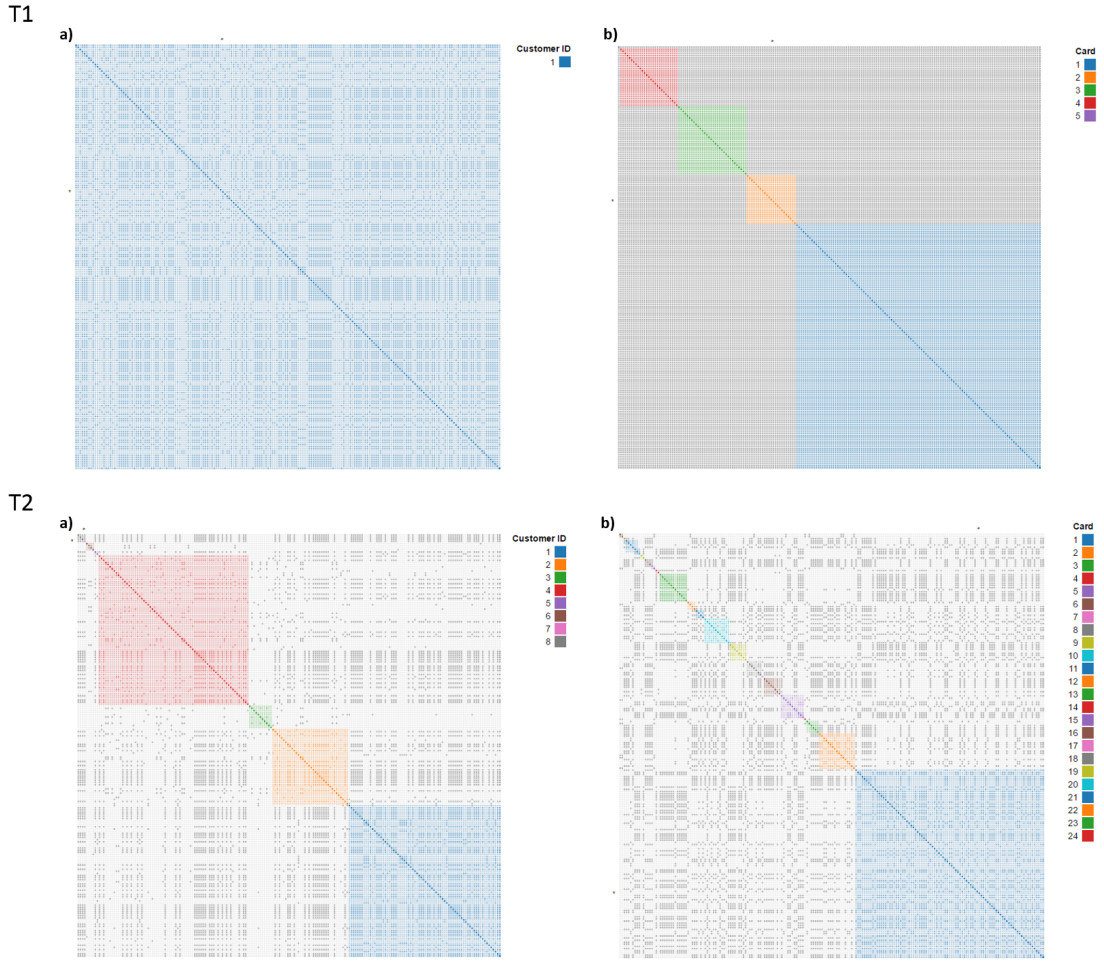

Figure 55: Matrix diagram representing transactions related with T1 (on top) and T2 (on the bottom) by credit card and related by customer ID and credit card. Diagram a) is using the option group by Customer ID, while in b) transactions are grouped by Card.

B.2.3 Force Directed Graph

To make the Force Directed Graph visualization more perceptible, parameters were adjust differently while analysing T1 and T2.

On transaction T1, the weight of the Card relation was increased to 5, in order to reinforce relationships using the same Card. Therefore, five node clusters become easily distinguishable on the graph. Figure 56, on the top, compares the graph with nodes coloured by Customer ID and by Card. It is evident that all transactions were originated by the same Customer ID and that 5 different cards were used.

To analyse transaction T2, the weight of the parameters was inverted, in order to reinforce relationships with the same Customer ID. Therefore, we attributed a weight of 5 to the parameter Customer ID and a weight of 1 to the parameter Card. Several clusters become visible, with even smaller clusters inside.

Comparing the resulting graph coloured by Customer ID and by Card (figure 56, on the bottom), it is noticeable that the same customer is using several cards and every card is being used by several customers and that some of the cards are being shared by some of the customers, which is consistent with a fraud pattern.

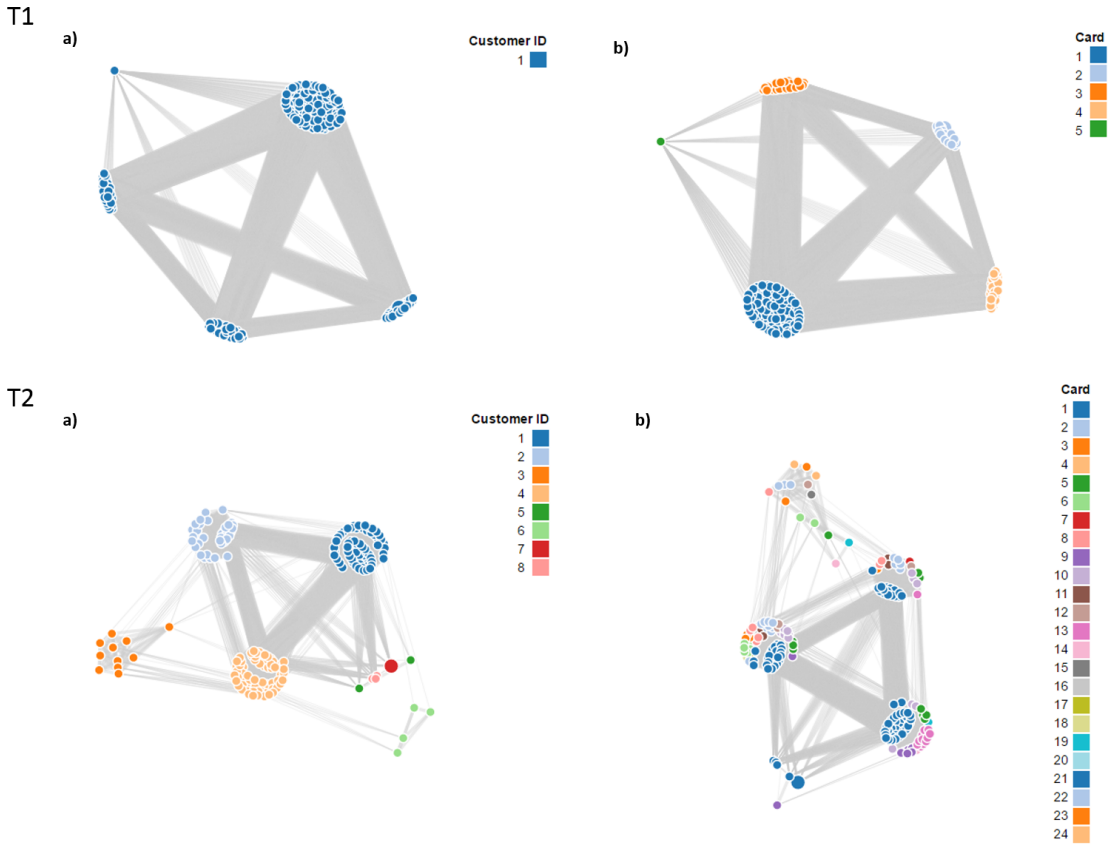

Figure 56: Force Directed Graph of transactions related with T1 (on the top) and with T2 (on the bottom) by credit card and related by customer ID and credit card. Diagram a) is using the option colour by Customer ID, while in b) transactions are coloured by Card.

B.2.4 Circular Diagram

Circular diagram was built using a two level hierarchy: nodes are distributed by Customer ID and, inside a Customer, by Card. Ligations between nodes represent relationships by Customer ID or Card.

On transaction T1 circular diagram (figure 57, on the top), 5 clusters are distinguishable: they correspond to the 5 cards being used for the same customer. When

the diagram is coloured by Customer ID all transactions share the same colour. Coloured by Card, each cluster displays a different colour.

Transaction T2 circular diagram (figure 57, on the bottom) is harder to interpret. Coloured by Customer ID, we can identify 8 different colour clusters, divided in smaller clusters. When coloured by Card, those smaller clusters represent the cards used by each customer. It becomes evident that each customer used several cards and that those cards were shared between different customer, which is consistent with a fraudulent pattern.

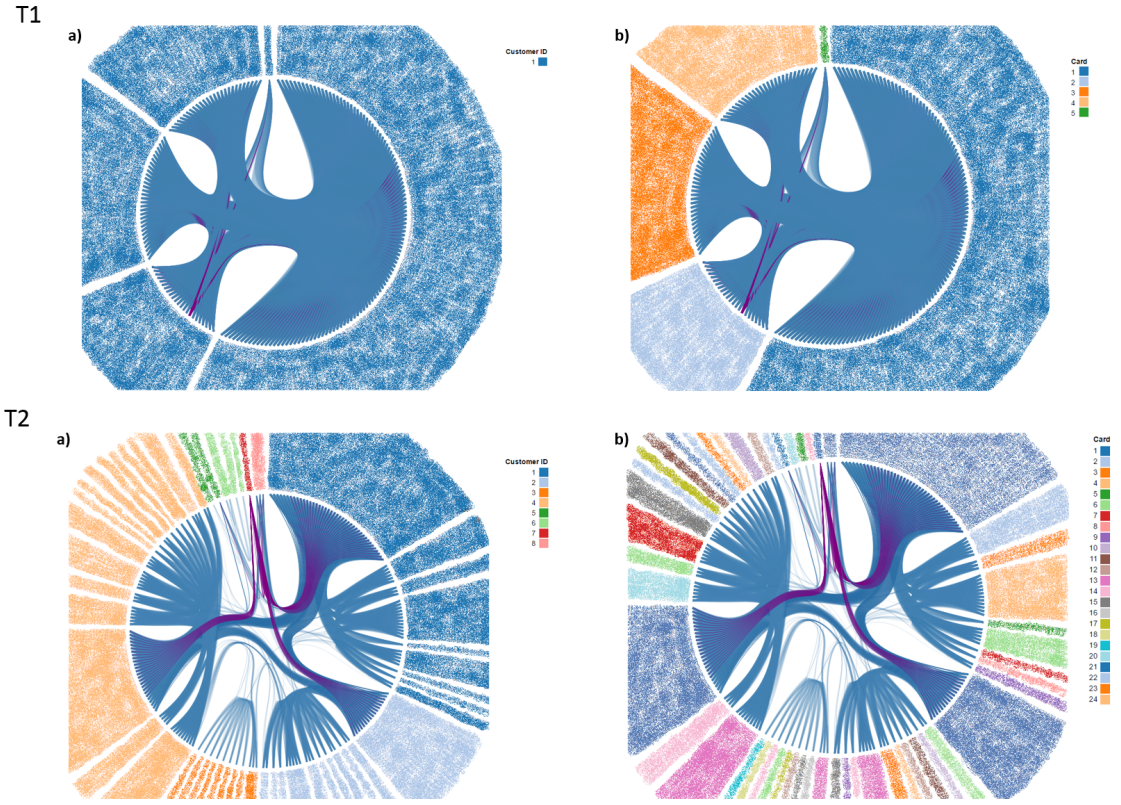

Figure 57: Circular Diagram of transactions related with T1 (on the top) and with T2 (on the bottom) by credit card and related by customer ID and credit card, hierarchized by Customer ID and by Card.
Diagram a) is using the option colour by Customer ID, while in b) transactions are coloured by Card. Transaction IDs were blurred out of the diagram, for confidentiality reasons.

B.2.5 Chord Diagram

On chord diagram, transactions are group by entities and relationships between those entities are represented as ribbons. This analysis compared the group option by Customer ID and by Card.

Transaction T1 grouped by Customer ID presents a single customer: all transactions were made using the same customer ID. When grouped by Card, 8 entities are

distinguished and all are linked with each other. Since the relationships displayed by the ribbons are due the sharing of the same Customer ID or/and the same Card, we can conclude that each different Card is sharing the parameter Customer ID.

Chord diagram of transaction T2 is more complex to analyse, since there are more Customer IDs and card involved in the same number of transactions. Eight different customer can be identified, sharing cards between themselves, as it is observed when the diagram is coloured by Customer ID. Coloured by Card it is distinguishable that 24 cards are being used and shared.

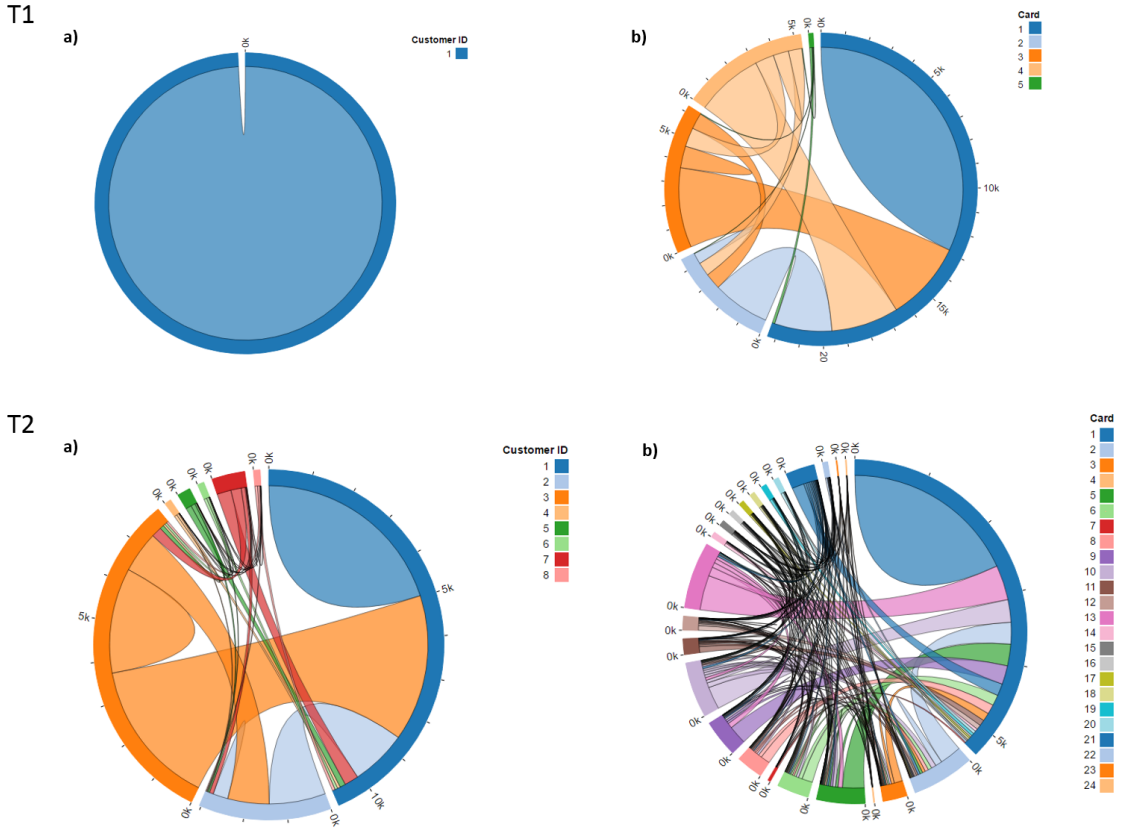

Figure 58: Chord Diagram of transactions related with T1 (on the top) and with T2 (on the bottom) by credit card and related by customer ID and credit card. Diagram a) is using the option colour by Customer ID, while in b) transactions are coloured by Card.

C Test Plan

. The purpose of this test is to evaluate different data visualizations and their capacity of being used to detect fraud. In each visualization, several transactions will be analyzed in order to detect fraud cases.

The following annex details the test plan for the user experience evaluation process.

C.1 Usability Testing

The test will take place in a controlled environment (meeting room) with an expected number of 8 users. We will provide the material (laptop, mouse and questionnaires) for the test. The expected time for the test will be 45 minutes per person.

A small introduction of the test duration and content will be made at the beginning of the test. It is important to stress out that we are evaluating the product and not the user ability to interact with it. After that, the user will be invited to do a small personal presentation (name, age, and job).

Since the capacity of interacting with the platform is not the main goal of the test, a brief demonstration of the platform will be made, with a test case of three related transactions. Then, the user will be asked to perform guided tasks drawing their own conclusions about the results obtained. In the end, the user will answer some questions about the experience and submit a SUS evaluation about the platform.

A screen recorder tool will be used to record the session, to get some measurements about tasks achievement and performance time.

The usability testing will consist of 3 main stages:

1. User tasks
2. Final Questions
3. SUS evaluation

C.1.1 User tasks

In this stage, the user will be invited to perform a series of tasks and to retrieve information from the visualizations, to evaluate the content displayed in the platform. Ideally the user should be comfortable to think out loud through the process and the tester should not give feedback or answer any questions.

TASK 1
1) Select visualization type "TABLE"
2) Select transaction (a)
3) Select all transactions using the same: Customer ID; Card; IP Address
4) Get transactions related by: Customer ID; IP Address

5) Interpret the results

TASK 2
6) Select visualization type "MATRIX"
7) Compare "Group By" options "Customer ID" and "IP Address"
8) Interpret the results

TASK 3
9) Select visualization type "FORCE DIRECTED"
10) Select transaction (b)
11) Get transactions related by: IP Address
12) Compare "Color Options" options "Customer ID" and "IP Address"
13) Interpret the results

TASK 4
14) Select visualization type "CIRCULAR"
15) Get transactions related by Card
16) Compare "Color Options" options "Customer ID" and "Card"
17) Interpret the results.

TASK 5
18) Select visualization type "FORCE DIRECTED"
19) Select transaction (c)
20) Increase number of maximum transactions to 200
21) Compare "Color Options" options "Customer ID" and "Card"
22) Interpret the results

TASK 6
23) Add relation parameter "Customer ID"
24) Increase weight of "Customer ID" parameter to 5
25) Compare color options "Card" and "Customer ID"
26) Interpret the results

TASK 7
27) Select visualization type "CIRCULAR"
28) Compare "Color Options" options "Customer ID" and "CARD"
29) Interpret the results

TASK 8
30) Select visualization type "MATRIX"
31) Compare "Group By" options "Customer ID" and "CARD"
32) Interpret the results

C.1.2 Final Questions

After the user ends all the required tasks, he will be asked to answer close and open-ended questions. Afterwards, the user will be invited to point out 2 negatives and 2 positives aspects of the experience and the application UI.

1. Which data visualization was your favorite and why?
2. What could make it better?
3. Point out 2 positives and 2 negatives aspects of this experience.
4. Do you want to add anything or make any question?

C.1.3 System Usability Scale

The System Usability Scale [60] provides a "quick and dirty", reliable tool for measuring the usability of a system. It consists of a 10 item questionnaire with five response options for respondents, from Strongly agree to Strongly disagree.

To avoid the natural tendency to provide a neutral answer, this questionnaire was modified, in order to have an even number of points in the scale. Therefore, an additional option for respondents was included.

The printed test will be provided to the user and he will evaluate the system by scoring the 10 statements below depending on where he stands for each specific statement.

	The System Usability Scale Standard Version	Strongly disagree					Strongly agree
		1	2	3	4	5	6
1	I think that I would like to use this system.	O	O	O	O	O	O
2	I found the system unnecessarily complex.	O	O	O	O	O	O
3	I thought the system was easy to use.	O	O	O	O	O	O
4	I think that I would need the support of a technical person to be able to use this system.	O	O	O	O	O	O
5	I found the various functions in the system were well integrated.	O	O	O	O	O	O
6	I thought there was too much inconsistency in this system.	O	O	O	O	O	O
7	I would imagine that most people would learn to use this system very quickly.	O	O	O	O	O	O
8	I found the system very cumbersome to use.	O	O	O	O	O	O
9	I felt very confident using the system.	O	O	O	O	O	O
10	I needed to learn a lot of things before I could get going with this system.	O	O	O	O	O	O

Figure 59: System Usability Scale

C.2 Tools and resources

The tools and resources needed for this test are:

- 1 Meeting room
- 2 Testers
- 8 Users
- 1 Laptop with access to the application and with a screen recorder tool installed
- 1 Mouse
- 8 Printed tests
- 2 Pens

C.3 Users

Users will be selected from Feedzai employees and must be selected in order to constitute a representative sample of users, as much as possible.

C.4 Results and Metrics

Test results and metrics will be extracted from the test, such as the average time to perform a specific task and the SUS scale, which will give us a very pragmatic usability evaluation.

We will also extract some qualitative data in the final questions.