Visualization and Self-Organising Maps for the Characterisation of Bank Clients

Catarina Maçãs*, Evgheni Polisciuc, and Penousal Machado

University of Coimbra, Centre for Informatics
and Systems of the University of Coimbra,
Department of Informatics Engineering, Portugal
{cmacas,evgheni,machado}@dei.uc.pt
http://www.cdv.dei.uc.pt

Abstract. The analysis and detection of fraudulent patterns in banking transactions are of most importance. However, it can be a laborious and time-consuming task. We propose a visualization tool—VaBank—to ease the analysis of banking transactions over time and enhance detection of the transactions' topology and suspicious behaviours. To reduce the visualization space, we apply a time matrix that aggregates the transactions by time and amount values. Additionally, to provide a mechanism to characterises the different sub-sets of transactions and facilitate the distinction between common and uncommon transactions, we represent the transactions' topology through matrix and force-directed layouts. More specifically, we present: (i) a visual tool for the analysis of bank transactions; (ii) the characterisation of the transactions' topology through a self-organising algorithm; (iii) the visual representation of each transaction through a glyph technique; and (iv) the assessment of the tool effectiveness and efficiency through a user study and usage scenario.

Keywords: Information Visualization, Glyph, Finance, Profiling, Self-Organising Maps

1 Introduction

The analysis of financial data and the search for fraudulent activities may prevent possible future losses for institutions and their clients, and for this reason, it is a task of high importance. The management of fraud usually focuses on three main pillars: (i) detection, defined by a continuous monitoring system that measures and evaluates possible fraudulent activities; (ii) prevention, defined by a preventive method that creates barriers to fraud; and (iii) response, defined

* This work is funded by national funds through the FCT—Foundation for Science and Technology, I.P., within the scope of the project CISUC—UID/CEC/00326/2020 and by European Social Fund, through the Regional Operational Program Centro 2020. The first author is funded by the Portuguese Foundation for Science and Technology (FCT), under the grant SFRH/BD/129481/2017 and by the project PS0738—LEXIA, under the grant 679746.

by a set of protocols that should be applied when fraud is detected [1]. In this work, we focus on the first pillar and develop a visualization tool that aims to facilitate the analysis of financial data and the detection of fraud.

Nowadays, experts in charge of fraud management support their analysis on tabular data, usually presented in the form of a spreadsheet and seldom supplemented with simple visualizations. With those methods, the inspection of irregularities and suspicious behaviours can be laborious, time-consuming, and arduous. Regarding the data to be analysed, it can be in a raw state or be the result of a previous analysis from Machine Learning (ML) systems trained to detect fraudulent behaviours. In both cases, experts' current tools may be of little use for the analysis and overview of such complex data. Additionally, as technology evolves and the techniques applied to detect fraud become publicly available, fraudsters adapt and modify their ways of acting [2]. This may prevent Machine Learning (ML) models from correctly detecting all fraudulent transactions and may lead to their incorrect classification. For this reason, investing only in ML algorithms for the detection of fraud can lead to undetected fraud cases.

To solve the aforementioned problems, especially the lack of tools to analyse both raw and pre-processed data, Information Visualization can be applied. Through visualization models that emphasise data patterns, it is possible to make fraud detection more reliable, effective, and efficient [2, 3]. Also, through the combination of computational means with our visual cognitive intelligence [4] and by enabling the detailed analysis of each suspicious behaviour that still needs to be carefully investigated, visualization can facilitate the analysis of financial data and reveal new undetected fraudulent patterns.

The visualization tool developed in this work is the result of a collaboration with Feedzai[1]. Feedzai is a world-leading company specialised in fraud prevention that owns a risk management platform powered by big data and ML. This platform is used mainly to identify fraudulent payment transactions and minimise risk in the financial industry (e.g., retail merchants and bank institutions). With their platform, Feedzai provides its clients with the possibility to analyse information and keep their customers' data and transactions safe. Also, Feedzai has its own fraud analysts whose goal is to provide a more detailed and humanised analysis of the data and detect unknown patterns of fraud. In this context, Feedzai highlights the need to design systems that can take advantage of the complementarity between humans and machines, to surpass the current limitations of humans and machines, while being provably beneficial [5, 6].

Feedzai's main goal is to provide its analysts with a visualization tool that enables the proper analysis and characterisation of different subsets of data—containing transactions made by several clients of a specific bank [2]. The analysts'

[1] Feedzai (`https://feedzai.com`) is the market leader in fighting financial crime with Artificial Intelligence. One of their main products is an advanced risk management platform.

[2] Note that due to the high sensitivity of the dataset, it was previously anonymised and encrypted, but retained the fraud patterns of the real datasets. This enables us

level of expertise in Information Visualization is reduced and their experience in analysing fraud may vary. During their analysis process, there is a lack of tools to aid them. In fact, they only have at their disposal spreadsheet-style tools and a limited data analysis platform, which makes the analysis of fraud an overwhelming task. In this matter, the creation of our visualization tool—*VaBank*—is intended to hasten their analysis process by giving information about data relations, such as time intervals between events, similarity, and recurring patterns, and to provide an overall sense of scale in the financial time-oriented data. In the end, these findings may lead to the detection of suspicious behaviours and the finding of fraudulent activities, enabling the banks to take action.

More specifically, with VaBank, we aim to ease the analysis of the distribution of bank transactions over time, the detection of the main characteristics and topology of those transactions and, with this, aid in the detection of suspicious behaviours. These objectives were based on the goals of Feedzai's fraud analysts, which can be summarised as: (i) be able to inspect collections of transactions in a single place—usually, these collections are grouped by attributes, such as client ID or location IP; (ii) understand the overall behaviours of a set of transactions; and (iii) detect the most common types of transactions.

Our tool is implemented in Java and uses Processing[3] to render the visualization. VaBank is divided into two main areas, the visualization of the transactions' distribution over time and the visualization of the client's topology of transactions. In the latter, we apply a Self-Organising Map (SOM) algorithm to represent the topology of a subset of transactions, enable the detection of the most common type of transactions, and with this, characterise the client main behaviours. Other visual representations for multi-dimensional data were not considered as we aimed to create a coherent visual representation of the transactions between the two main areas described above. For example, if a parallel-coordinates were to be used, the visualization in the first main area would augment in complexity, as multiple parallel-coordinates would be needed to represent every transaction at the different periods of time. The challenge to create a single representation led us to the definition of a glyph simple enough to be perceived at first glance, but with another level of detail for a more thorough analysis. Also, SOMs have already proved its usefulness and robustness for the analysis of large amounts of data [7]. The visualization of their results provide a visual summary of the data topology and can ease the interpretation of behaviours in a single image [8, 9]. We present the SOM's results through two visualization techniques: a matrix and force-directed projections. Both aim to represent the profiling of a group of transactions and enable the understanding of the characteristics of the most common transactions. Finally, the transaction history visualization provides a set of analytical features, enabling the analyst to navigate, explore, and analyse the sequence of transactions over time.

to visually explore the data in real case scenarios, without compromising the users' anonymity.

[3] Processing is an open-source graphical library

Our main contributions are (i) a user-centred visual tool, developed with the aid of fraud experts; (ii) a method that characterises the topology of the transaction through a SOM algorithm; (iii) the visual characterisation of transactions through complex glyphs; and (iv) a usage scenario and a user study that assess the tool effectiveness. Based on the analysts' feedback, we could conclude that our tool can improve substantially their line of work which currently involves the time-consuming analysis of spreadsheets.

The current article extends the work presented in [10] in two main aspects. First, we give a more detailed description of the dataset and its processing. Also, we extend our description of the tasks, the VaBank design, and how we performed the user testing with experts in fraud analysis. Second, we expand the validation of the VaBank tool with three usage scenarios. The usage scenarios goal is to highlight the efficiency and effectiveness of VaBank in enabling a detailed analysis of the transactions.

The remainder of the article is structured as follows. In Section 2, we present the related work on fraud visualization and self-organising algorithms. In Section 3, we introduce the dataset and how it was processed. In Section 4, we present the tasks and requirements of our tool and, in Section 5, we give a detailed description of the VaBank design. In Section 6, we describe three different usage scenarios and, in Section 7, we present a user testing with experts in fraud analysis. In Section 8, we discuss the results of the tests. Finally, in Section 9, we present our conclusions.

2 Related Work

The visual exploration of data has already proved its value concerning exploratory data analysis, as the user is directly involved in the exploration process, adjusting its goals along the analysis [11]. In this section we present the related work on the visualization of fraud in the finance domain. Additionally, we present the related work on Self-Organising Maps.

2.1 Visualization of Fraud in Finance

In what concerns the visual highlight of fraudulent activities, and regardless of the domain of application, the most common visualization techniques are line and bar charts and node-link diagrams. These techniques are used to represent changes over time, facilitate the comparison of categorical values, and represent networks and relationships, respectively. Focusing only on the financial domain, two surveys present a set of projects which apply techniques, such as parallel coordinate plots, scatterplots, and bar and line charts [12, 13]. For a more detailed description of the techniques and the different taxonomies, please refer to the works [12, 14].

For the representation of specific financial fraud patterns, six works can be found concerning the visualization of (i) *Stock Market Fraud*, which focus on the analysis of abnormal changes in stock market values along time [15, 16]; (ii)

Profile Analysis, which focus on the analysis of personal bank transactions [17]; (iii) *Credit Card Fraud*, which focus on the analysis of improper use of credit cards [18]; and (iv) *Money Laundering*, which focus on the analysis of the network of transactions [19, 20]. From these, four projects [17, 18, 19, 20] focus merely on the improvement of the respective automatic evaluation systems, not applying visualization for the manual analysis of fraud cases. Also, in the work of Sakoda et al. [18], they visualise directly the fraud labels given by the ML system, not giving further details of each transaction to enhance its analysis. Finally, from this subset, most tools apply more than one visualization technique in separate or multiple views.

From our research, we only found one visualization model related to the visualization of bank data. Wire Viz [21], is a coordinated visualization tool that aims to identify specific keywords within a set of transactions. Also, they apply different views to depict relationships over time. For example, they use a keyword network to represent relationships among keywords, a heatmap to show the relationships among accounts and keywords, and a time-series line chart to represent the transactions over time. Their goals are to give an overview of the data, provide the ability to aggregate and organise groups of transactions, and compare individual records [21].

With this research, we could conclude that the analysis of fraudulent activities through visualization is gaining popularity, but its use to detect specific types of fraud is uncommon. In the case of bank transactions, the only related work uses a different type of dataset, which contains transactions to and from other banks, whereas, in the dataset made available by Feedzai, we only have access to the transactions made from the accounts of a specific bank. With this dataset, we are not able to follow the connections between different transactions, being our main aim to characterise the transactions of specific clients that may be referred to as suspicious cases. We argue that to properly understand the behaviours of a certain client, a more detailed analysis of his/her patterns of transactions must be conducted, so it is possible to distinguish atypical and suspicious transactions from the common transactions.

2.2 Self-Organising Maps

Self-Organising Map (SOM) take advantage of artificial neural networks to map high-dimensional data onto a discretised low-dimensional grid [22]. Therefore, SOM is a method for dimensionality reduction that preserves topological and metric relationships of the input data. SOMs are a powerful tool for communicating complex, nonlinear relationships among high-dimensional data through simple graphical representations. Although there are multiple variants, the traditional SOM passes through different stages that affect the state of the network [22]. In the first, all neurons are initialised with random values. Then, for each datum of the training data input, the so-called Best Matching Unit (BMU) is defined. This is done by computing Euclidean distances to all the neurons and choosing the closest one. Finally, the weights of the BMU and the neighbour neurons are adjusted towards the input data, according to a Gaussian function—which shrinks with time. This process is then repeated for each input vector for a predefined number of cycles.

Since the present work deals with mixed data, we present SOM algorithms which work with that type of data. The topological self-organising algorithm for analysing mixed variables was proposed in [23], in which categorical data is encoded to binary variables. Also, the algorithm uses variable weights to adjust the relevance of each feature in the data. Hsu et al. [24, 25] proposed another method in which they use semantics between attributes to encode the distance hierarchy measure for categorical data. Similarly, the authors in [26] use semantic similarity inherent in categorical data to describe distance hierarchy by a value representation scheme. The authors in [27] use distance hierarchies to unify categorical and numerical values, and measure the distances in those hierarchies. Finally, in [28] a frequency-based distance measure was used for categorical data and a traditional Euclidean distance for continuous values.

Visualization of Self-Organising Maps The visualization of SOMs is typically concerned with the projection of neurons into a 2D/3D grid. The most common projection is the Unified Distance Matrix (U-Matrix), in which neurons are placed in a grid and the Euclidean distances between neighbouring neurons are represented through a greyscale colour palette. This visual mapping can be used in the detection of clusters [29, 30] or in the definition of thresholds [31]. Additionally, hexagonal grids [32] can also be used [33], increasing neighbourhood relations, although not always resulting in more detailed insights [33]. The results of SOMs have also been used as data inputs for other visualization models. In most cases, researchers used SOMs to define clusters or characterise different behaviours and then represent such groups in the visualization models. In [34], a 3D SOM was used to define clusters categorised visually with colour, which later is applied in geographic areas with different characteristics. In [35], SOM was also used to define clusters in data, and then those clusters were represented through various visualization models, such as parallel coordinates and Chernoff faces. In fact, the usage of Chernoff faces and glyphs, in general, was found in multiple works, which will be discussed in more detail later. Finally, in [36], the clusters resulting from the SOM algorithm were visualised through a two views visualization, consisting on the representation of the clusters on a map and in a temporal grid.

To improve the reading and understanding of each neuron, some works used more complex glyphs. In [37], the neurons are represented through a timeline, portraying the temporal profile of call logs, and, in the background, a circle is drawn with the size depending on the number of elements used to train each neuron. In [38], each neuron is represented by a squared glyph coloured according to the quantisation error and, inside each square, a line is drawn to represent a certain trajectory. In [39], the neurons are represented with a radar glyph which shows the consumption value of a specific product. Finally, in [40] a rose diagram is applied to represent the weights of each feature of the SOM.

Self-Organising Maps in Finance The application of SOM algorithms to analyse transactional data have been applied in a variety of projects. The majority of them apply SOMs to provide an analytical view on the financial market trajectories [41, 38, 42] and to analyse their stability and monitor multi-dimensional financial data [43]. Other works applied SOM to better comprehend the stock market dynamics[44] or to analyse the financial performance of companies [7].

3 Data Analysis and Preprocessing

We worked with an anonymised dataset that contains only the transactions generated by the clients of a certain bank—there is no data about the transactions that each client received. Each transaction of the dataset is characterised by attributes corresponding to the: client (e.g., ID, IBAN), location (e.g.,Client IP, Country IP), monetary amount (e.g., amount, currency), transaction (e.g., type, descriptor, fraud label), beneficiary details (e.g., IBAN), and date. Each transaction can be of two types: online, corresponding to regular transactions; and business, corresponding to business transactions. All clients can have transactions of both types. The transactions also have a descriptor, composed of two or three acronyms, that characterise the transaction according to (i) the interface used; (ii) the type of operation (e.g., national, international, loan); and (iii) whether it is for a new beneficiary or not. These characteristics must be known by the analysts, so they can be properly analysed. This task has a high level of difficulty as these descriptors can have different combinations. To enable a better understanding of the descriptor elements, we herein list them according to each type of transaction:

Business Transactions:

- Type of Interface: ATM Specific, Telephone, ATM, and Branch
- Type of Operations: Cash In and National
- Type of Beneficiary: New and Old.

Online Transactions:

- Type of Interface: Barc. Mobile, MBWay, Web, and App
- Type of Operations: Instant, International, National, Loan, Address change, and Agenda
- Type of Beneficiary: New and Old

Additionally, all transactions are labelled by the bank as fraudulent or not. For this project, we group dynamically the transactions of a certain subset in different range scales in two axes: time and monetary amount. In terms of time granularity, the time axis can be divided in different ranges of days, being one day the smallest granule possible. Also, to be able to properly summarise the data, for each pair $[time, amount]$ we aggregate the transactions with the same characteristics (i.e., the same values for the attributes type, descriptor, and fraud label).

3.1 SOM Algorithm

We applied a variant of the Frequency Neuron Mixed Self-Organising Map (FMSOM), a SOM algorithm prepared to handle mixed data [28]. It consists of preserving the original algorithm for handling the numerical variables and extending the neuron prototype with a set of category frequency vectors. The algorithm follows the traditional *competition, cooperation* and *adaptation* process. Since we focus on the visualization tier of the SOM and not on the algorithm, any other method could be used. However, the FMSOM model allowed us to adapt it to define the dissimilarity between neurons, used in the visualization of the transaction's topology.

Features First, we extracted the features for each input raw data. In our project, 7 features and their types were identified: *amount*, *day* of week, *month* of the year, *year*, *time passed* since the last and until the next transactions (in milliseconds), *fraud*, *transaction type*, *operation type*, *beneficiary*, and *interface channel*. The later five features were briefly described in Section 3 and cannot be fully revealed due to the specificity and sensitivity of the dataset. The *amount* is the amount of money involved in the transaction. From the date of a transaction, we extract only the day of week $[1-7]$, the month of the year $[1-12]$, and the year. The features *time passed* since the last transaction and until the next transaction are previously calculated and are intended to capture the patterns of the transactional regularity.

Dissimilarity Metric We applied different measures to compute the distances between neurons. We applied the traditional Euclidean distance for continuous values and the measure based on probabilities (described in [28]) for categorical features. Ultimately, two types of dissimilarity measures were defined: one for the training of the SOM; another for the visualization.

Regarding the SOM domain, as in FMSOM [28], the dissimilarity measure between the neuron and the input feature vector consists of the following. Suppose that P is the number of input feature vectors $X_p = [x_{p1}, ..., x_{pF}]$, where F is the number of features in that vector. Also, suppose that n and k are the number of continuous and categorical features, respectively, where $[a_k^1, ..., a_k^r]$ is the set of categories of the k_{th} feature. Finally, suppose that the reference vector of the i_{th} neuron is $W_i = [W_{i1}, ..., W_{in}, W_{in+1}, ..., W_{iK}]$, where I is the number of the neurons in the network. With that said, the dissimilarity between an input vector and the reference vector of a neuron is defined as the sum of the numerical and categorical parts. The numerical part is calculated using Euclidean distance on normalised values. For the categorical dissimilarity measure the sum of the partial dissimilarities is calculated, i.e., the dissimilarity is measured as the probability of the reference vector not containing the category in the input vector. For more details on the FMSOM algorithm consult [28].

Regarding the visualization domain, the dissimilarity measure between two neurons is determined as follows. For the numerical part, the traditional Euclidean distance is applied $Dn(W_i, W_j) = \sqrt{\sum_{z=1}^{n}(W_{iz} - W_{jz})^2}$. For the categorical features the dissimilarity measure was defined as the Euclidean distance between the probabilities for each of the categories present in the reference vector $Dk(W_i, W_j) = \sqrt{\sum_{z=n}^{k}\sum_{m=1}^{r}(W_{iz}[a^m] - W_{jz}[a^m])}$. So, the final dissimilarity measure is given by $d(W_i, W_j) = Dn(W_i, W_j) + Dk(W_i, W_j)$.

4 Tasks and Requirements

From our collaboration with the fraud detection company, we were able to hold several meetings with their analysts, which aided us to better define the domain and requirements for the analysis of the bank data. The analysts emphasised two main tasks: [**T1**] comprehend of the transaction history; and [**T2**] detect of the most common types of transactions. The latter is especially important as it enables the distinction between typical and atypical behaviours.

The analysts described their line of work, referring that their analysis usually starts by grouping the data by a specific attribute. Usually, they group the transactions by a specific client ID to better analyse and characterise the client's transactions. Then, they search for groups of transactions with similar characteristics, especially the ones labelled as fraud. This task is especially difficult using a spreadsheet, as if the transactions are not ordered, common attributes will not stand out. From all attributes, the analysts referred to the amount spent, type of transaction, and fraud label, as the most relevant. In the end, they referred to the importance of detecting similar transactions and identifying the profile of the client or a subset of transactions. Through our meetings, the analysts defined five requirements to which VaBank should comply:

R1 **Search by field.** The analysts usually sort the data by a certain field, such as client IBAN, client ID, or Country of IP, and analyse the transactions with common values on those fields. The creation of a mechanism that enables the analyst to easily select a field and choose a certain value of that field to group the transactions is of utmost importance. This will speed up the analysis process and ease the analysis of all transactions with common values;

R2 **Distinguish amount values.** When dealing with bank transactions, the transacted amount can be a sign of fraudulent activity, being transactions with high amounts, or above a certain threshold worth of a more detailed analysis. The visual sorting of the transactions by their amounts can enhance the detection of suspicious transactions;

R3 **Distinguish transactions.** By visually characterising each transaction, the analysts can more easily distinguish the transactions and focus their attention on the ones of the same type. With this, they can perceive the behaviours within the different types of transactions, facilitating the detection of atypical behaviours;

R4 **Search common fields.** When dealing with this data through spreadsheets, the analysts have difficulties in detecting transactions that share more than

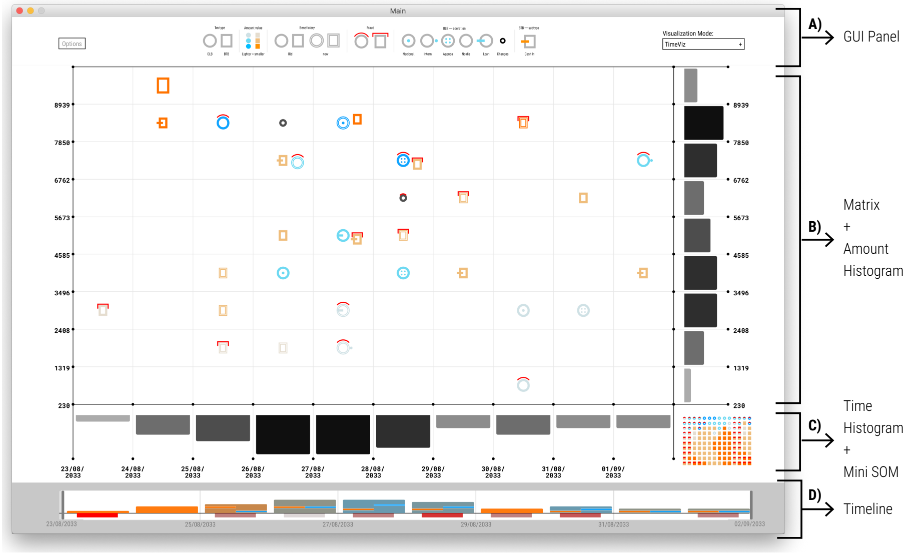


Fig. 1. Transaction History view and its components, from top to bottom: GUI Panel (A); Matrix View and Amount Histogram (B); Time Histogram and Mini SOM (C); and Timeline (D).

one attribute. This is of utmost importance when analysing fraudulent transactions which can share attributes with others. For this reason, it is important to implement a mechanism that enables the analyst to select an attribute and highlight all transactions with that same attribute.

R5 Detect typical transactions. Understanding the most common types of transactions can enhance the analysis of the data and aid the analyst in the detection of unusual transactions, which can be related to fraudulent behaviours. Hence, it is important to characterise the space and facilitate the detection of typical transactions in a certain subset of the data;

5 VaBank Design

The tool is divided into three views: the *Transactions History*; the *Transactions Topology*; and the *Transactions Relationships*. The first view (Figure 1) aims to answer the task **[T1]** and arranges all transactions by time and amount. The last two views aim to answer to the task **[T2]** and display the results of the SOM algorithm (see Section 3.1) in a grid and through a force-directed graph, respectively. With these views, for the same subset of the data, we aim to enable the analysis of the transactions by time (i.e., first view), and enable the analysis of their topology (i.e., second and third views). All three views share a common visual element, the transactions. We developed a glyph that serves to identify the type of transaction and its position in time. With this, we aim to facilitate the

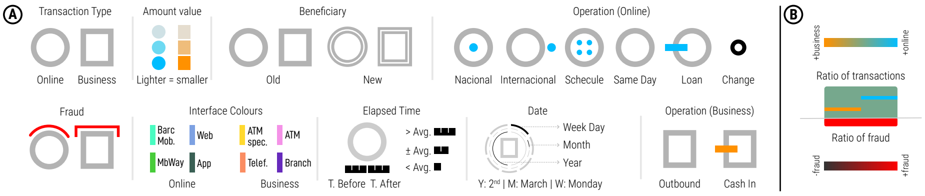


Fig. 2. Glyph elements that characterise each transaction (side A) and timeline bar composition and respective colour ranges (side B).

distinction between transactions with different characteristics and to provide coherence between views. In the following subsections, we present the design rationale of the glyph and the three views.

5.1 Transaction Glyph

To ease the distinction and visual characterisation of the transactions, we implemented a glyph [**R3**]. The glyphs are composed of three levels of visual detail. These levels were defined together with the company's analysts, according to the relevance of the types of attributes when analysing bank data. First, the analysts aimed to distinguish online transactions from business transactions. Then, the transaction amount and whether it was considered as fraud or not are analysed. These three characteristics represent the first level of visual impact. Then, the analysts want to drill down and distinguish between: inbound and outbound transactions; and new and old beneficiaries. These characteristics represent the second level of visual impact. Finally, the time characterisation of each transaction and the interface with which the transaction was made were defined as less important than the described above. For this reason, they are grouped into the third level of visual impact and should have a lower visual impact.

As colour has a high impact on visualization [45], we apply colour to emphasise the characteristics of the first visual level. We apply different hues to the types of transaction: orange for business; blue for online. Additionally, we use different shapes to emphasise this distinction between transaction types—a rectangle for business; and a circle for online. Then, we use saturation to represent the amount: the brighter the colour, the higher the amount. As small differences in saturation would be imperceptible to the human eye, we defined three levels of saturation to distinguish: low, medium, and high amounts. These levels are computed as follows. We compute the average amount $\overline{x}$, define a window w, and if the value is: below $\overline{x} - w$, we consider the amount as low; between $\overline{x} - w$ and $\overline{x} + w$, the amount is medium; and higher than $\overline{x} + w$, the value is high. The window w is a percentage of the average value that was defined in collaboration with the company's analyst. Finally, to represent a fraudulent transaction, we place a red line above the main shape (see Figure 2, A). Note that these colours were not tested on colour blind people.

The transactions' shape is complemented with a set of symbols that represent the types of operation. They are divided according to the directionality of the transaction, outbound or inbound [4]. The inbound is represented by the same symbol in online (i.e., Loan) and business (i.e., Cash In) transactions: a vertically centred horizontal rectangle positioned on the left. The outbound operations are represented as depicted in Figure 2, in which the business transaction only have one type, and the online transactions have five. As the new beneficiary characteristic is a binary value, we represent transactions for new beneficiaries by dividing the stroke of the main shape in two. If the beneficiary is not new, no change is made (Figure 2, A).

For the third level, we represent the year, month, and day of the week of the transaction. Each time variable is represented by a ring with a different radius centred in the main shape, being the year the smallest ring, and the day of the week the biggest ring (Figure 2). To distinguish periods of time, we divide the ring into 7 wedges, for the days of the week; 12 wedges for the months; and, for the years, in the total number of years in the dataset. All wedges are coloured in light grey, except the wedge that marks the period of the transaction, coloured in black. The day of the week has a thicker stroke, as the analysts referred it is the most important time variable. We also represent the elapsed time between the current transaction and the previous and following transactions. We apply an equal rationale to represent these two-time distances. As with the amount thresholds, we defined three levels of time distances that are computed in the same way. These three levels are represented as depicted in Figure 2. Note that for the sake of simplicity this data was aggregated, even though in the SOM we use absolute values. Finally, the interface of the transaction is represented by filling the elapsed time's shape with the corresponding interface colour (Figure 2, A).

The glyphs used in the views concerning the SOM's result make use of all representations described above. However, in the Transaction History view, we only represent the first two levels of visual detail, as time is already being represented in the x-axis.

5.2 Transaction History View

In this view, there are a set of visualization models that display different data aggregations. The main representation, which occupies more canvas space, is the *Transaction Matrix* (Figure 1, B). It divides the space in different ranges of monetary values on the y-axis and temporal values on the x-axis [**R2**]. The transactions' glyphs are then distributed by the cells of the matrix, according to their date and amount. If more than one transaction with the same characteristics (defined in 5.1) occurs within the same cell, they are aggregated and its glyph grows in size. The placement within each cell is made through a circle

[4] Note that the inbound are transactions made only by the client, when asking for a loan (in online transactions) or when doing a deposit (in business transactions)

packing algorithm which starts by placing the biggest glyph in the middle of the cell and the others around it.

In the bottom and right sides of the Transaction Matrix, histograms are drawn to show the total number of transactions per column and row, respectively (Figure 1, B and C). The histogram's bars are coloured according to the number of transactions: the darker the bar, the higher the number of transactions. Also, in the bottom right corner of the Transaction Matrix area, we draw a small matrix of glyphs that represents the result of a SOM algorithm, concerning three attributes: amount, transaction type, and fraud (Figure 1, C). With this, we aim to give a visual hint to the analyst about the distribution of the different transactions, enhancing the understanding of typical/atypical transactions [**R5**].

At the bottom of the canvas, we placed an interactive timeline, so the analyst can select different periods of time (Figure 1, D). This timeline represents all time-span. To be able to represent all data in the timeline, we applied a hierarchical time aggregation algorithm that aggregates semantically the transactions according to the space of the timeline (see Section 5.2). The timeline is divided horizontally into equal sections, representing different periods of time with the same duration. Each section of the timeline is vertically divided into two parts.

In the upper part, we represent the number of transactions through a bar. To put it briefly, each bar is drawn as follows: (i) its height represents the total number of transactions; and (ii) its main colour is defined by a gradient between blue and orange—the bluer, the higher the number of online transactions, the more orange, the higher the number of business transactions (Figure 2, B). With this, we aim to represent which type of transaction occurs the most. To give a more detailed view, we also represent the quantity of each transaction type with two thin rectangles which are drawn inside the previous bar. They are placed horizontally according to the type of transaction—being the business one on the left and the online one on the right—and are placed vertically according to the percentage of occurrence. Additionally, they are coloured according to the transaction type.

In the bottom part of each timeline section, we place a rectangle with a predefined height. This rectangle is only visible if one or more fraudulent transactions occur. Then, it is coloured according to the percentage of fraudulent transactions in that specific period of time. The higher the number of fraudulent transactions, the brighter and redder the bar will be (Figure 2, B). If no fraud occurs, no bar is drawn.

Hierarchical Temporal Aggregation Fixed timelines can create multiple problems (see for example [46, 47]). For example, different time spans can result in either a tremendously cluttered timeline; a timeline with an uneven distribution of the time bars (e.g., one bar on the left and the other bars concentrated on the right); or a timeline that uses inefficiently the canvas space, due to the time granularity (e.g., one thin bar on the left and another on the right). With our algorithm, we intend to solve the problem of fixed timelines. The main goal is to allow the representation of any temporal range where the timeline would adapt its granularity and adjust the size of time bars.

Our adaptive timeline algorithm takes as arguments the available space for the timeline and the minimal width of a time bar. The algorithm follows an iterative top-down approach. We start at the biggest time unit existing in the computation systems (e.g., epoch), and descent, iterating over consecutive ISO time units (e.g., years, quarter years, months) until we find an optimal balance between the time granularity and the size of the time bars. The algorithm has to meet one single criterion that is tested at each temporal resolution. Consider T_i being the time tier currently evaluated, T_{min} and T_{max} being the minimal and maximal timestamp of the selected data subset, W_{min} being the minimal allowed width for the bars, and W_{total} being the width of the timeline. So, the criteria to determine the time resolution and the width of a bar is computed as follows: $W_{total}/T_{i+1}(T_{max} - T_{min}) < W_{min}$.

Note that we compute the width of bars at the $i + 1$ temporal tier. If the bar width at the next tier is smaller than W_{min} we stop, and the current tier is the one that we are looking for. The left part of the expression is the found width of bars.

Interaction To enable the analysts to analyse the transactions in more detail, we defined a simple set of interaction techniques. In the Transaction Matrix, the analyst can hover each glyph to see more details—Country IP, amount, beneficiary, and the number of transactions. If the analyst clicks on a glyph, these details are fixed in the canvas. By doing so, the analyst can interact with each one of the attributes. If the analyst clicks on an attribute, the transactions which share that same attribute will be highlighted with a black ring. With this, we aim to enhance the understanding of the transactions that may share the same suspicious attributes [**R4**]. Also, the user can interact with each bar of the histograms. By hovering a bar, the total number of transactions is shown and the analysts can more easily perceive the total number of transactions in a certain period of time (x-axis) or the total number of transactions within a certain range of monetary values (y-axis).

We also defined a set of interaction techniques for the timeline. The analyst can select different periods of time to visualise in the transaction matrix. To do so, the analyst must drag two vertical bars which are positioned in the leftmost and rightmost parts of the timeline area. By selecting a shorter period of time, the transaction matrix will be more detailed (i.e., the different periods of time in the x-axis will have shorter durations, being one day the shortest possible). With this, we aim to enable the analysts to see in more detail the distribution of the transactions over time. Also, the analyst can drag the selected to maintain the selected duration, but change the initial and final periods of time. Finally, the analyst can hover each bar of the timeline. By doing so, a set of statistics are made available concerning the total number of transaction in that period of time, the start and end dates, the percentage of online and business transactions, and the percentage of fraud.

5.3 Transaction Topology View

In this view, we visualise the result of the SOM algorithm defined in 3.1. The SOM algorithm uses all transactions available of a predefined bank client. To visualise its result, we use the positions of the neurons in the SOMs matrix to distribute the glyphs on the canvas within a grid with the same number of columns and rows (Figure 3, top). Also, and as referred previously, we use the three levels of visual detail to represent each neuron (see Section 5.1).

This approach enables the analyst to visualise the most common types of transactions through the analysis of the distribution of the different glyphs (representing the transaction's characteristics) in the matrix [5]. However, this view lacks a more detailed representation of the dataset, which could enable, for example, the representation of how many transactions are related to each neuron and which neuron is the most representative of the dataset. The latter task is especially difficult to achieve when more than one feature is being represented in the glyphs, as it can hinder the comparison between glyphs. To overcome this, we implemented a second approach, in which we place each neuron within a force-directed graph and represent their relations to the transactions. With this, we aim to achieve a better understanding of the client's profile.

5.4 Transaction Relations View

For the force-directed graph, neurons and sets of transactions are represented as nodes and are positioned within the canvas according to their dissimilarity measure: the similar two neurons are, the closer they will get (Figure 3, bottom). The force-directed graph can be seen as a simplification of the SOM result produced and visualised in the Transaction Topology View. Our implementation of the graph is based on the *Force Atlas 2* algorithm [48]. All nodes have forces of repulsion towards each other so they do not overlap. However, only nodes whose dissimilarity is below a predefined threshold have forces of attraction. This makes similar nodes to get closer to each other, generating clusters defined by the SOM topology. Additionally, we added a gravitational force that pulls all nodes towards the centre of the canvas. The higher the number of connections between nodes, the higher the gravitational force. With this, clusters which are more representative of the dataset will be in the centre of the canvas, and the ones representing atypical transactions in the periphery.

To avoid clutter, only neurons selected as BMU in the training process of the SOM are represented. We opted to filter the neurons with this method, as the neurons that are selected as BMU are the ones which are more similar to the transactions within the dataset, and for this reason, are the ones which are more representative. Also, the transactions which have the same neuron as BMU are aggregated and this aggregation is represented with a node. These new nodes' forces of attraction are defined by their average force of attraction to other neurons.

[5] Note that, as in the analysis of any SOM, the number of glyphs in the canvas is not representative of the number of transactions within the dataset

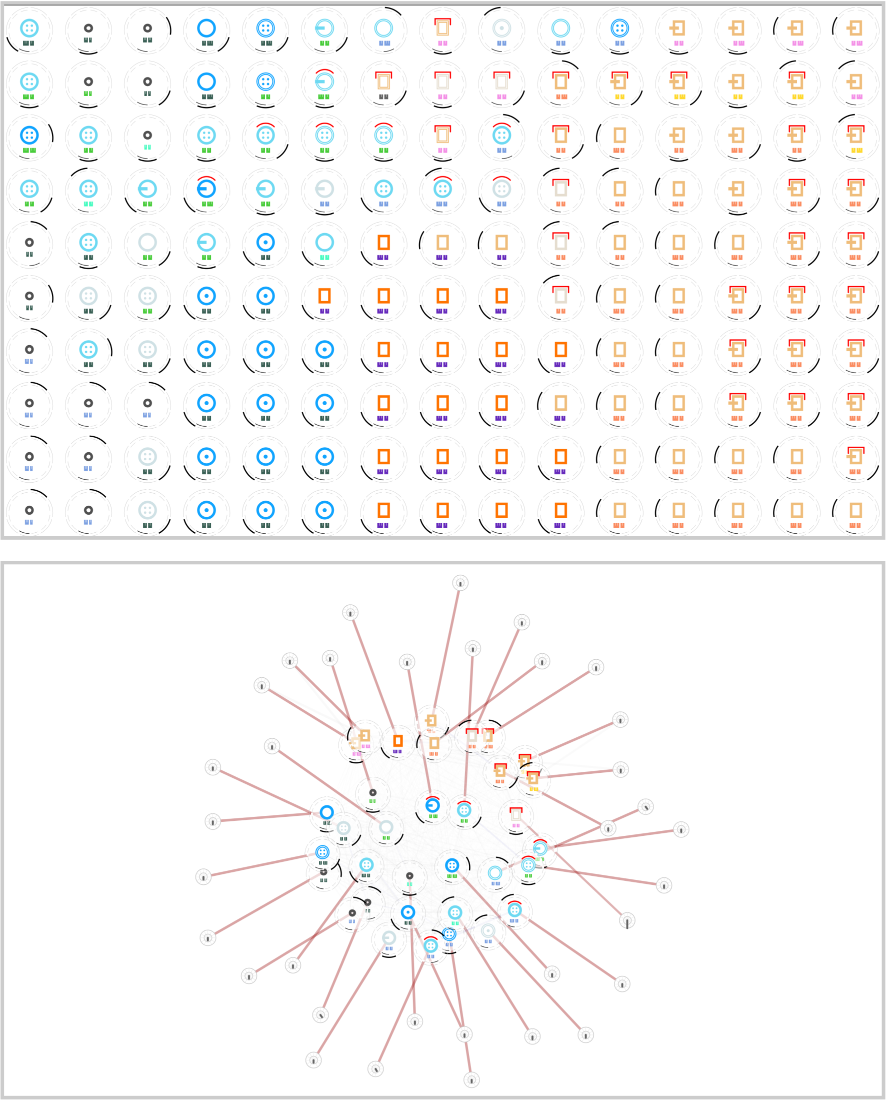


Fig. 3. Projections of the SOM results for the same bank client through the matrix projection (top) and force-directed graph (bottom). The self-organising map used in the matrix projection is generated with all transactions available from a certain client. The force-directed graph can be seen as a simplification of the matrix, as it only visualises the most representative neurons from the self-organising algorithm used in the matrix projection. Both projections aim to represent the most common transactions of a specific bank client and, therefore, characterise the client.

The nodes have distinct representations. The neurons are represented with the glyphs described in 5.1. For the groups of transactions, we use a circular chart that represents the number of transactions by month of occurrence. This representation is intentionally simpler since our main goal is to give more visual impact to the result of the SOM. Also, if these nodes are connected to a certain neuron, it means they share similar characteristics, being redundant to use the glyphs approach.

We used lines to connect the nodes. These lines are coloured: (i) in red if they connect a node representing a group of transactions and their BMU neuron; (ii) in light grey, if they connect a group of transactions and other neurons which are also similar to them, but are not their BMU; and (iii) in blue, if they connect two similar neurons. These lines are represented to enhance the comprehension of the proximity of the nodes, but as they should have less visual emphasis, their opacity and thickness diminish according to the similarity values.

5.5 Control Panel

To enable a better transition between views we created a *Control Panel* (Figure 1, A). By clicking on the "Options" button, on the upper left corner, the *Options Panel* is shown, containing a list of all unique attributes of a predefined field—client ID. This list is scrollable and is sorted in an ascending way, according to the number of transactions of each client. Also, each row contains a set of statistics concerning the grouped transactions: the total number of transactions, the maximum, minimum, and average amount values, and the percentage of fraudulent transactions. On the *Options Panel*, the analyst can also access a list of fields and select a different one to group the transactions **[R1]**. On the upper right corner of the *Control Panel*, there is a dropdown that enables the analyst to change between the three views. Finally, in the middle of the *Control Panel*, a caption is shown to describe the glyphs that represent the transaction's characteristics (Figure 1, A). This caption is especially important due to the complexity of the glyphs: with it, the analyst can easily read the glyph without needing to memorise or search for the caption anywhere else.

6 Usage Scenario

In this section, we discuss three usage scenarios in which we analyse subsets of the dataset with fraudulent transactions. With this, we aim to highlight the efficiency and effectiveness of VaBank in enabling a detailed analysis of the data. In each scenario, we visualise the transactions made by a certain bank client in one month [6]. Due to the limited time range, all scenarios present a reduced number of transactions. However, we argue that this is not a limitation as our model is prepared to aggregate the data in different time ranges, and for this reason, a larger dataset would not add more difficulty to the analysis. Also,

[6] this small temporal range is due to the limited accessibility to the data

the number of transactions per period of time would not change significantly, meaning that wider time spans would only result in bigger time ranges in the timeline. Nonetheless, with our timeline, the user can select smaller periods of time to reduce the time span being represented in the Transaction Matrix, enabling a more detailed analysis of each transaction.

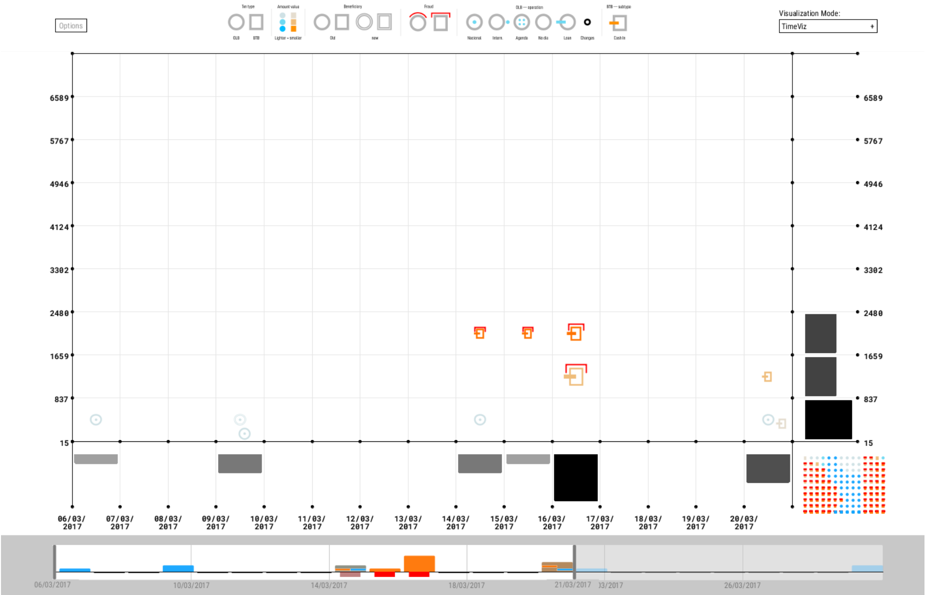
6.1 Client A

In Figure 4 (a), we can see the *Transactions History* View of *Client A*. We can instantly perceive, through the positioning of the first transactions, that the monetary value of those transactions is relatively low—concerning the other ranges of values visible in the y-axis. Also, by looking at the bar chart in the timeline, it is possible to understand that the transactions tend to occur periodically—there is an initial set of transactions, then no transactions are made in the following three days, then another set of transactions are made, and so on. On March 14, there was a business transaction with higher amount values that was marked by the bank as fraud. We can also see that Client A tried consecutively to make that type of transactions on the two following days with similar and smaller values, but got the same result, a fraud label by the bank. All of these transactions are Cash In operations, which means that the client attempted to add money to his/her account. Later, on March 2$^{\text{nd}}$, we can see the same type of transactions with smaller values, however, this time they were not labelled as fraud. By looking at the small matrix—generated from the SOM—it is possible to see that the majority of the business transactions were considered fraudulent, especially the ones with high values.

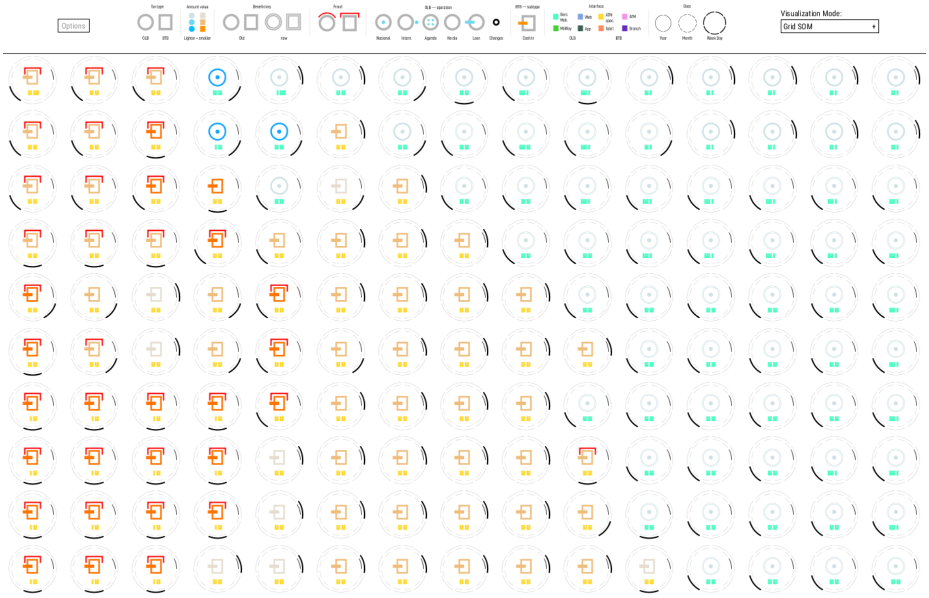
When analysing the *Transactions Topology View* (Figure 4 (b)), we can verify the assumptions made previously and see that for the business transactions Client A used mainly the ATM interface (yellow) and for the online transactions the interface used was the Barc. Mobile. Also, we can see that the majority of the online transactions were of the national type and for new beneficiaries. Finally, by checking the *Transactions Relations View*, we can see these clear distinctions between online and business transactions (Figure 5). In the cluster of business transactions, we can easily define two sub-clusters: the fraudulent transactions with high values and the ones with smaller values. Also, it is possible to see that the business transactions with low values were made on Tuesday, whereas the fraudulent ones occurred between Wednesday and Friday. For all these reasons, this client can be seen as suspicious.

6.2 Client B

In the second usage scenario, there were also visible fraudulent transactions (Figure 6 (a)). Although the values are low, in comparison to the previous client, this client attempted several transactions of the business type with different amounts and aimed to add money to the account. When comparing these business transactions with the rest of Client B transactions, which are usually placed below the €50 limit, the business transactions are of high value. Through the timeline,



(a) Transaction History View. With this first view, it is possible to detect a group of business transactions



(b) Transactions Topology View. Through this view it is possible to perceive that the majority of the business transactions with higher values are considered as fraudulent.

Fig. 4. Two different views of Client A.

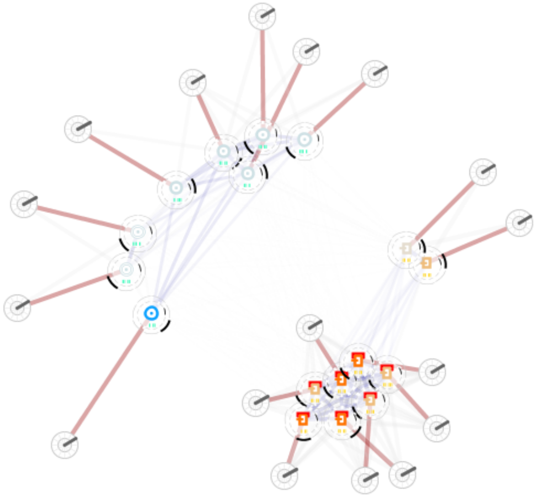


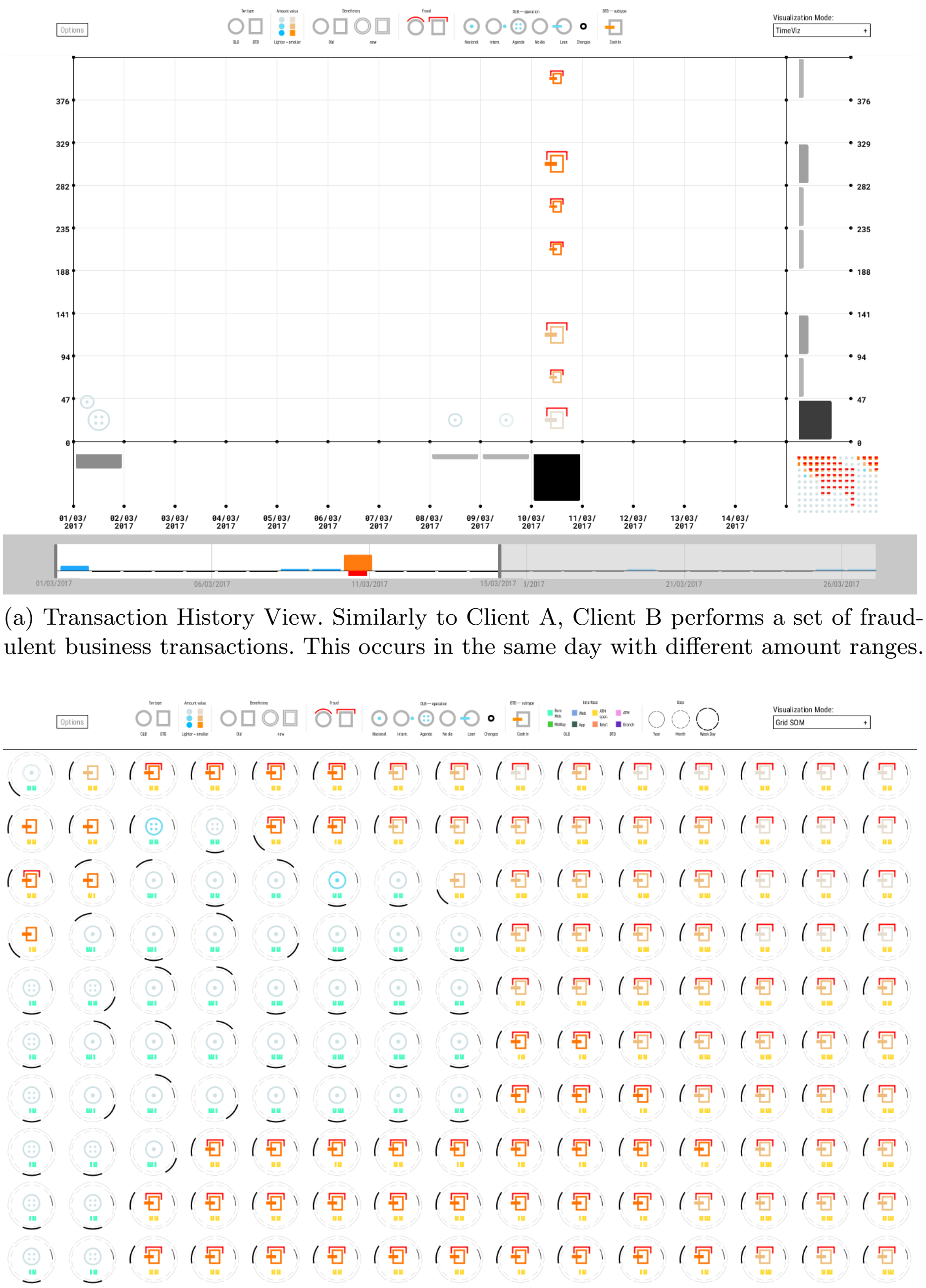
Fig. 5. Transaction Relation View of Client A. Two major clusters can be seen, separating online from business transactions. Also, the business cluster is subdivided into fraudulent and non-fraudulent transactions.

we can see that this client, after the peak of business transactions on March 10, made far fewer transactions. By looking at the small SOM matrix, we can see that this client behaves similarly to the previous one, as the majority of the business transactions are considered fraudulent and the online transactions are all of reduced value.

All the previous assertions can be verified with the *Transactions Topology View* (Figure 6 (b)). The business transactions are, in the majority of the cases, considered as fraud, and the online transactions have in their majority smaller value ranges—in comparison to the business transactions. Additionally, the online transactions are divided into two subtypes: the ones of the agenda type and the others of the national type. With the aid of the *Transactions Relations View*, we can easily visualise these assumptions through the two well-defined clusters, one for the online transactions and the other for the business transactions (Figure 7). Similarly to Client A, this client can be seen as suspicious.

6.3 Client C

When analysing the third client's data, we can see that it differs from the previous examples as the majority of the transactions are of the online type (Figure 8 (a)). The values in these transactions fall in their majority in two different ranges: in the lowest range, from €30 to €500, and in the highest range, above €4200. Also in both ranges, there are fraudulent transactions of the online type. The fraudulent transactions are common in higher values, but not so common in



(a) Transaction History View. Similarly to Client A, Client B performs a set of fraudulent business transactions. This occurs in the same day with different amount ranges.

(b) Transaction Topology View. The majority of the neurons are of the business type and are considered to be fraudulent.

Fig. 6. Two different views of Client B.

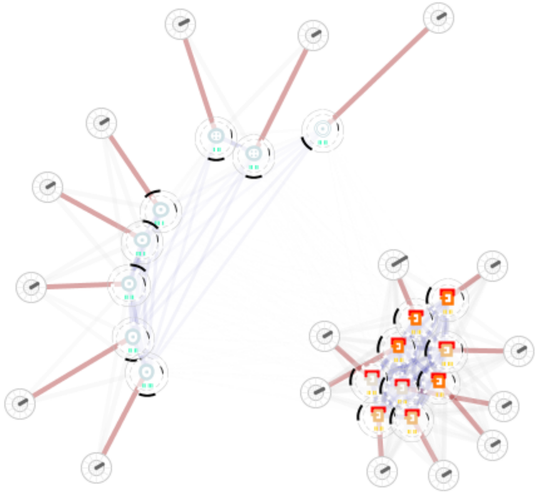
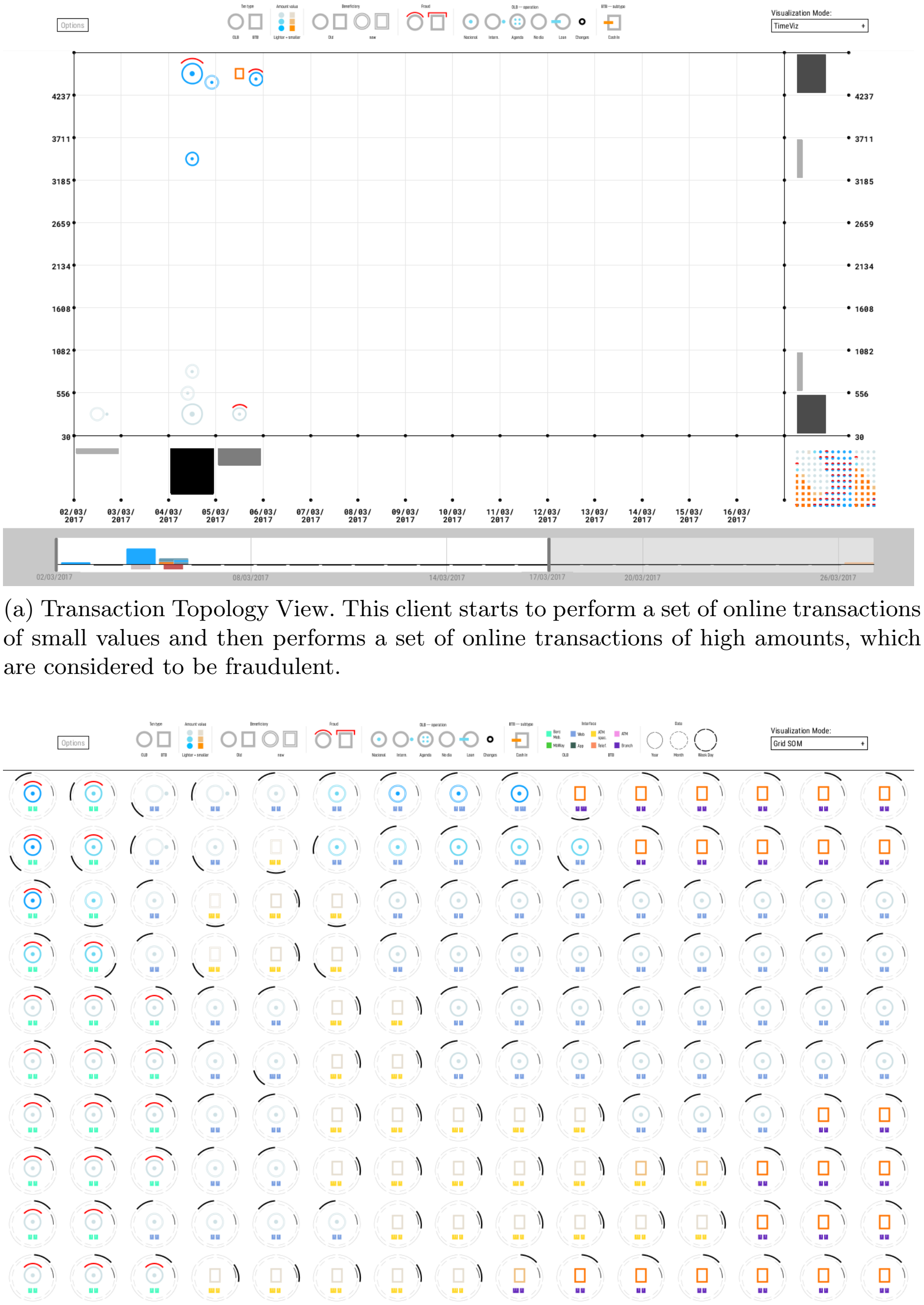


Fig. 7. Transaction Relations View of Client B. In this visualization, two clusters can be found, one of the fraudulent business type and the other of online transactions of small amounts.

lower ranges. In this case, we can also see that, on March 5, there are fraudulent transactions in both ranges, which is uncommon. This client starts by doing an international transaction of low value and on the other day makes a few more national transactions to new beneficiaries of both high and low values, being only detected two fraudulent transactions out of eight. On the next day, there is a high amount of business transactions and two national transactions. This can be defined as a suspicious behaviour, since there are no more transactions in the following days, until March 26. By looking at the small SOM matrix, we can see a more varied SOM representation, in which fraud appears only on the national online transactions.

By looking at the *Transactions Topology View*, it is possible to see that the majority of the transactions have low-value ranges (Figure 8 (b)). Also, it is interesting to see that fraudulent transactions are made via Barc. Mobile and non-fraudulent national transactions are made via the web. This may indicate a breach in one of the applications and should be analysed in more detail. Also, it is possible to perceive that the business transactions of low ranges were made via ATM and the transactions with high values via Branch.

When analysing the *Transactions Relations View*, we can see three main clusters and two outliers—which are the business transactions of low and high ranges (Figure 9). Also, in online transactions, we can see a distinction between fraudulent and non-fraudulent transactions. Among the non-fraudulent, we can distinguish four types of transactions: international, to new beneficiaries, national with low values, and national with high values. From this, we can refer



(a) Transaction Topology View. This client starts to perform a set of online transactions of small values and then performs a set of online transactions of high amounts, which are considered to be fraudulent.

(b) Transaction Topology View. In this view, we can see a more varied transaction matrix, when comparing with Client A and Client B. This means that this client performs a more varied types of transactions.

Fig. 8. Two Different Views of Client C.

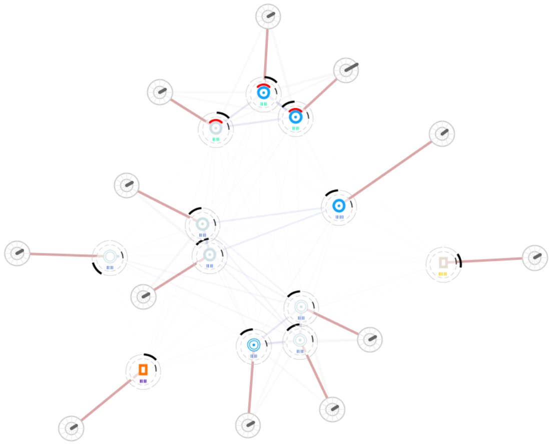


Fig. 9. Transaction Relations View of Client C. In this visualization, it is possible to perceive two types of transactions which can be considered as outliers, the transactions of the business type oh small and high amounts. Also, it is possible to distinguish two types of online transactions, the fraudulent, and the non-fraudulent.

that with the graph representation of the SOM's results, we were able to analyse more rapidly the different transactions and their relations.

7 User Testing

To evaluate the tool's usefulness and effectiveness in the analysis of bank transactions, we performed user tests with a group of fraud analysts from the Feedzai company that were not present during the tool development. In this user testing, the participants were asked to (i) perform a set of specific tasks, (ii) to analyse the transactions from two clients through the interaction with the VaBank tool, and (iii) to give feedback on the aesthetics, interpretability, aid, and learning curve of each one of the three views. The tasks were defined to validate the models and determine the effectiveness of the visual encodings. The second part—the analysis of a subset of transactions—was defined to assess the complete functionality of the VaBank tool as a whole and whether the analysts were able to retrieve insights from the visualizations, proving is usefulness in the analysis of bank transactions. The third part was defined so the opinions of the analysts could be registered and analysed.

7.1 Participants

The user testing was performed by five fraud analysts. These analysts worked for Feedzai, but have no a-priori knowledge about the VaBank tool. On average, they worked in fraud analysis for five years, being the participant with the least years of experience, been working for three years, and the one with the most years of experience, been working with fraud for eight years. Also, three of the analysts had no experience with working with Information Visualization, and the other two had a reduced number of interactions with the field. Despite this being a reduced number of participants, this user testing aimed at understanding the impact of a tool such as VaBank in the analysis process of fraud experts—which are more used to deal with spreadsheets. For this reason, we believe that this number of participants was sufficient to fulfil the test requirements and provide a general sense of the VaBank impact on their analysis process.

7.2 Methodology

The tests were performed as follows: (i) we introduced the glyphs of the transactions, the views of the tool, and respective interaction mechanisms; (ii) we asked the analysts to perform 18 tasks concerning: the *Transactions History View* (6), the interpretability of the glyphs (4), the *Transactions Topology View* (4), and the *Transactions Relations View* (4); (iii) then, the analysts analysed two clients in terms of fraudulent behaviours; and (iv) the analysts were asked to give feedback on the models concerning aesthetics, interpretability, aid in the analysis, and learning curve. The second and third part of the tests were timed and, at the end of each task or analysis, the analysts were asked to rate the difficulty of the exercise and certainty of their answers on a scale from 1 to 5—from low to high, respectively.

The 18 tasks of the user testing were divided into 4 groups, depending on the component they aim to validate: **G1** Transaction History view; **G2** Transaction glyphs; **G3** SOM Matrix; and **G4** SOM Graph. In the Transactions History View, we tested the analysts' ability to comprehend temporal patterns and the transactions' distribution concerning time and amount values. In the views related to the SOM projections, we aimed to compare both views and perceive which one was more useful and efficient in solving tasks like counting clusters and identifying all glyphs from a certain attribute. For this reason, the tasks are equal for both views.

The third part of the test—which is concerned with the interaction with the VaBank tool and the analysis of two different clients' data—aims to understand the tool's usefulness and its ability in aiding the analysts to detect suspicious patterns and possible frauds. During the performance of this part of the test, the analysts were asked to explore and analyse the visualization, explain out loud what they were seeing at each moment of their exploration, and refer to whether the client was fraudulent, non-fraudulent, or suspicious.

The final part of the test was also intended to give to the analysts the opportunity to express their opinions on the tool. Although such feedback might be subjective, it is an indicator of the tool's impact within the analysts' workflow and can give clues on its effectiveness and efficiency.

All tests occurred in the same room within the Feedzai installations and were performed under the same conditions (i.e., the participants had access to

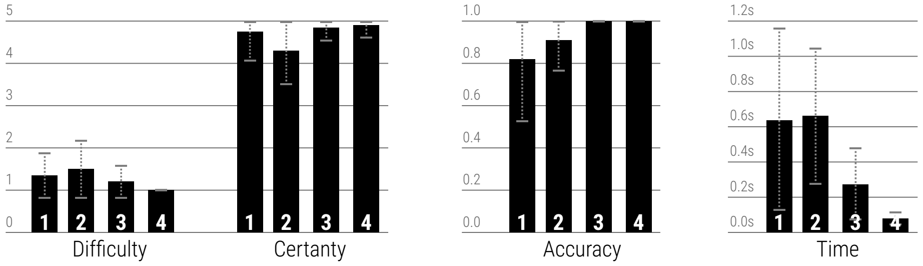


Fig. 10. Difficulty, certainty, accuracy and time values for the 4 tasks groups. In general, the difficulty of the tasks was considered to be low, and the certainty and accuracy are considered to be high. With regard to time, the majority of the tasks was completed in less than one minute.

the same computer and performed the test in the same sequence). We recorded the audio from each test so we could analyse each session afterwards.

7.3 Results

In Figure 10, we summarise the results concerning difficulty, certainty, accuracy, and duration for each group of tasks. Hereafter, we further analyse each group of tasks, discuss the results from the third part of the test, and analyse the analysts' feedback.

Tasks Analysis The tasks related to the analysis of the *Transaction History View* (**G1**) and glyphs (**G2**) were the ones which arouse more difficulty. Nonetheless, all values are low, considering that on average the difficulty was no higher than two (i.e., the second-lowest level of difficulty). Regarding the Transaction History View, the analysts had more difficulties in interpreting the positioning of the glyphs in the grid and the histograms. For example, for the task "In which period of time the business transactions had the highest amount?", some analysts started to look at the histogram on the right, which gives the total number of transactions for each range of amount values. However, as this was the first question of the test, they were still assimilating all the information and rationale of the tool. The analysts also had some difficulty in interpreting the glyphs, which made their certainty to be lower than the other groups. Nonetheless, the certainty on average was no lower than four (i.e., the second-highest level of certainty). Also, an interesting point is that the accuracy of the analysts' answers for the glyphs tasks is higher than the accuracy for the Transaction History View tasks. With this, we could perceive that, as the glyphs were complex, the analysts were not certain if they were characterising all their attributes correctly, which caused the lower rates of certainty. Nonetheless, in the majority of the tasks related to the glyphs, their answers were accurate.

The groups of tasks related to the SOM analysis—Transaction Topology View and Transaction Relations View—took less time to perform (20 seconds, on average), had 100% of accuracy, and were the ones in which the analysts had more certainty in their answers and less difficulty in completing the tasks. Comparing both views, the Transaction Relations View (**G4**) had the lowest duration and the difficulty of completion was also considered low. This can be explained by the fact that, as the graph is less complex (has fewer glyphs), for the same tasks the analysts could analyse more quickly the glyphs and their relationships. These good results on both views might also indicate the good acceptance of such models and the ease with which the analysts could interpret the topology of the transactions.

VaBank Analysis The third part of the test was concerned with the free exploration and analysis of the transactions of two clients. These clients have two different behaviours: Client A has a suspicious behaviour at the end of his/her data, and Client B commits fraud at the beginning of his/her data.

The majority of the analysts interacted with the tool in the same way. Hence, we hereafter summarise their interaction when analysing both clients—Client A and Client B. At the beginning of the Client A analysis, the analysts spotted no fraud in the first period range and all detected a weekly periodicity of online transactions of the agenda type with low values. One analyst started to be suspicious when perceived that there were 13 transactions to different new beneficiaries. Then, the analysts scrolled on the timeline to see the other transactions of that month. By doing so, the analysts understood that there was a disruption of the initial pattern. At the end of the month, the transactions had no pattern, being scattered along the last days, and the rate and value of the transactions increased. Through the interaction with the glyphs, one analyst noted that some transactions on the same day were made in different countries, which was considered suspicious. Also, by interacting with the glyphs, another analyst found that the beneficiary attribute changed in all transactions but the Internet Service Provider (ISP) was always the same. Additionally, after analysing the data through the Transaction Relations View and Transaction Topology View, one analyst referred that the suspicious behaviours were more evident in the Transaction History View than in the other views.

In summary, the majority of the analysts identified Client A as a suspicious case especially due to the pattern changes and the increase of amount and rate of transactions.

In regards Client B, the analysts directly detected the fraudulent activities through the glyphs. In this subset, the client asked for a loan of increased value that was considered as fraud and on the same day performed several online transactions of the agenda type, which were also considered as fraud. All analysts were intrigued by the fraudulent transactions, and all interacted with the glyph to try to understand what were the attributes of those transactions. By doing so, they could perceive that the transactions were made to different beneficiaries. The majority of the analysts referred to this type of behaviour as an external attack on a legitimate client account. Also, one analyst stated that it was also suspicious that Client B tried so many transactions of increased value, and then, after one week, made another transaction of a relatively small value. In summary, the majority of the analysts identified Client A as a suspicious case especially due to the pattern changes and the increase of amount and rate of transactions.

In summary, Client B was instantly classified as fraudulent, for his attempts of doing several transactions with high amounts for different accounts. Also, most analysts referred to Client B as an account that might have been hacked. As this client had few transactions, the analysts could see every transaction in the transaction matrix, without needing to interact with the timeline.

Feedback At the end of each test, the analysts rated each view in terms of aesthetics, interpretability, aid in the analysis, and learning curve. The Transaction History view got a higher rate in terms of aesthetics and aid. Additionally, it was defined as easier to interpret but had lower ratings in terms of the learning curve. This last rate may be caused by the complexity of this view, which included the histograms, the small SOM matrix, and the timeline.

Concerning the Transaction Topology View and the Transaction Relations View, the analysts took more time to complete the tasks with the first and rated it with higher values of difficulty. However, the Transaction Topology View was seen as a better aid for the analysis of the transaction patterns and was also defined as easier to learn, compared to the Transaction Relations View. In fact, the Transaction Relations View was the view with the lowest ratings in terms of aesthetic value and aid in the analysis of the data.

At the end of the tests, some analysts made some comments on the tool. They referred to the Transaction Topology View as a good auxiliary for their work and referred that with more practice the glyphs would get easier to read and interpret. One analyst also suggested a new positioning of the glyphs in the transaction matrix: to place them in each cell radially to represent the hours at which the transaction was made. At the end of each test, the analysts rated each view in terms of aesthetics, interpretability, aid in the analysis, and learning curve. The Transaction History view got a higher rate in terms of aesthetics and aid. Additionally, it was defined as easier to interpret but had lower ratings in terms of the learning curve. This last rate may be caused by the complexity of this view, which included the histograms, the small SOM matrix, and the timeline.

8 Discussion

Through our interaction with the fraud analysts, we were able to define the two main tasks to which VaBank should answer: to enable the visualization of the transactions over time and to enable their profile characterisation. The analysts also aided us in the definition of the specific requirements for the tool which allowed us to define the visualization models and interaction mechanisms. These first steps revealed to be important for the development of a tool to be used for the analysis and detection of suspicious behaviours in bank transactions.

Through the user testing, we could validate our tool in terms of efficiency, as most of the tasks were completed in reduced times—on average the tasks took less than 1.3 minutes to complete—and the exploration and analysis of the clients' data were also completed in a short time—took on average 4 minutes, which the analysts referred to as a good time for the analysis in comparison to their current tools. We could validate the tool in terms of effectiveness, as all analysts were able to complete correctly the tasks and also, through their interaction with VaBank, they were able to analyse the details of the transactions, their main characteristics, and detect suspicious behaviours.

With the analysis of the tasks' results, we could assess the interpretability of the visualization models. For example, we could understand that despite the complexity of the glyphs, the three levels of visual impact achieved their purpose, as the analysts could focus on the first level (the type of transaction, amount, and fraud) and with a more close reading analyse the operations types and the rest of the transaction's characteristics. Also, although during the execution of the tasks the Transaction History View was seen as the most difficult, after the interaction, the analysts found it to be easier to interact with. Also, through the analysts' feedback, we could understand that this view was well received by the analysts which defined it as a good auxiliary for their work. We could also perceive that, although the Transaction Relations View was faster to analyse, defined as easier to learn and in which the analysts were more certain about their answers, this view was also seen as less informative than the Transaction Topology View. The Transaction Topology View was seen by the analysts as a better aid for the analysis of the transaction patterns and was also defined as easier to learn.

With the analysis of the results of the second part, we could conclude that all analysts understood the tool's interaction mechanisms and all were able to interact properly with the tool. Also, the analysts took a small amount of time to analyse and perceive the types of behaviours of the analysed clients. With this, we can conclude that VaBank can aid in the detection of suspicious behaviours, which in turn, can improve the analysts' decisions.

Concerning the analysts' feedback, they stated that after the completion of the tasks they were more familiarised with the tool, and could easily use all interactive features. Additionally, they referred to the highlight of transactions with the same attribute as a good feature that was relevant for their line of work. This highlight aided in the creation of relationships between transactions and in the analysis of their attributes. Nonetheless, all visual elements were well received and understood. One analyst also referred that the timeline was an important asset as it enabled the visualization of different periods of time and the understanding of the types and amount of transactions on different time periods. Also, the representation and highlight of fraud were well understood by every analyst.

9 Conclusion

In this work, we explored different visual solutions for the representation of temporal patterns in the finance domain. We presented our design choices for VaBank, a visualization tool which aims to represent the typical behaviours and emphasise atypical ones in bank datasets. VaBank is a user-centred visualization tool developed in the context of a partnership with Feedzai—a Portuguese Fraud Detection Company—and intended to be implemented in the workflow of the company's fraud analysts. The company's main aim for the tool was that it could promote an efficient analysis of the dataset and could emphasise suspicious behaviours.

More specifically, we represented the temporal patterns of bank transaction data. With the collaboration with Feedzai, we were able to define the main requirements and tasks that would enable our tool to improve their analysis concerning their current tool—spreadsheets. Thus, our visualization models focus on: (i) the visual representation of the transaction characteristics through a glyph visualization; (ii) the temporal visualization of the transactions; (iii) the characterisation of the transactions topology through a Self-Organising Map (SOM) algorithm; and (iv) the projection of the SOM results into a matrix and a force-directed graph.

We validated and compared the different visualization components of the tool through formative and summative evaluations with experts in fraud detection. Through these tests, we could assess the effectiveness of the tool on the characterisation of the transactions. The analysts were able to properly analyse the visualization and detect different behaviours in different bank clients. In summary, the results showed that the tool was well received by the analysts and it could enhance their analysis, overpassing their current method—spreadsheets.

We contribute to the visualization domain in finance with a tool which focuses on the characterisation of bank transactions, on the representation of the topology of the transactions and, consequently, on the highlight of uncommon behaviours. By enabling in the same tool the visualization of the transactions along time—emphasising the ones with higher amounts—and their topology—emphasising the typical behaviours—, we were able to promote a better analysis of atypical transactions and suspicious behaviours. In conclusion, the presented work demonstrates that VaBank is effective and efficient for the analysis of bank data and in the detection of suspicious behaviours. Also, the characterisation of transactions with complex glyphs can aid in the understanding of transaction patterns and facilitate the analysis of the overall data.

References

[1] "Fraud Prevention, Detection, and Response". In: *Essentials of Forensic Accounting*. John Wiley & Sons, Ltd, 2017. Chap. 8, pp. 211–243. ISBN: 9781119449423. DOI: 10.1002/9781119449423.ch8.

[2] Richard J Bolton and David J Hand. "Statistical fraud detection: A review". In: *Statistical science* (2002), pp. 235–249.

[3] William N. Dilla and Robyn L. Raschke. "Data visualization for fraud detection: Practice implications and a call for future research". In: *International Journal of Accounting Information Systems* 16 (2015), pp. 1–22. ISSN: 1467-0895. DOI: https://doi.org/10.1016/j.accinf.2015.01.001. URL: http://www.sciencedirect.com/science/article/pii/S1467089515000020.

[4] Victoria L Lemieux et al. "Using visual analytics to enhance data exploration and knowledge discovery in financial systemic risk analysis: The multivariate density estimator". In: *iConference 2014 Proceedings* (2014).

[5] Stuart Russell. *Human compatible: Artificial intelligence and the problem of control*. Penguin, 2019.

[6] Melanie Mitchell. *Artificial intelligence: A guide for thinking humans*. Penguin UK, 2019.

[7] Tomas Eklund et al. "Assessing the Feasibility of Using Self-Organizing Maps for Data Mining Financial Information". In: *Proceedings of the 10th European Conference on Information Systems (ECIS) 2002*. Ed. by Stanislaw Wrycza. Vol. 1. AIS, 2002.

[8] Melody Y. Kiang and Ajith Kumar. "An Evaluation of Self-Organizing Map Networks as a Robust Alternative to Factor Analysis in Data Mining Applications". In: *Information Systems Research* 12.2 (2001), pp. 177–194. ISSN: 10477047, 15265536. URL: http://www.jstor.org/stable/23011078.

[9] A Costea et al. "Analyzing economical performance of central-east-European countries Using neural networks and cluster analysis". In: *Proceedings of the Fifth International Symposium on Economic Informatics*. Bucharest, Romania. 2001, pp. 1006–1011.

[10] Catarina Maçãs, Evgheni Polisciuc, and Penousal Machado. "VaBank: Visual Analytics for Banking Transactions". In: *24th International Conference Information Visualisation, IV 2020, Melbourne, Australia, September 7-11, 2020*. 2020, pp. 336–343. DOI: 10.1109/IV51561.2020.00062. URL: https://doi.org/10.1109/IV51561.2020.00062.

[11] Daniel A Keim. "Information visualization and visual data mining". In: *IEEE transactions on Visualization and Computer Graphics* 8.1 (2002), pp. 1–8.

[12] S. Ko et al. "A Survey on Visual Analysis Approaches for Financial Data". In: *Computer Graphics Forum* 35.3 (2016), pp. 599–617. DOI: https://doi.org/10.1111/cgf.12931. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12931. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12931.

[13] Maxime Dumas, Michael J McGuffin, and Victoria L Lemieux. "Financevis. net-a visual survey of financial data visualizations". In: *Poster Abstracts of IEEE Conference on Visualization*. Vol. 2. 2014.

[14] Roger A. Leite et al. "Visual analytics for event detection: Focusing on fraud". In: *Visual Informatics* 2.4 (2018), pp. 198–212. ISSN: 2468-502X.

DOI: https://doi.org/10.1016/j.visinf.2018.11.001. URL: http://www.sciencedirect.com/science/article/pii/S2468502X18300548.

[15] M. L. Huang, J. Liang, and Q. V. Nguyen. "A Visualization Approach for Frauds Detection in Financial Market". In: *2009 13th International Conference Information Visualisation.* 2009, pp. 197–202. DOI: 10.1109/IV.2009.23.

[16] J. Dale Kirkland et al. "The NASD Regulation Advanced-Detection System (ADS)". In: *AI Magazine* 20.1 (Mar. 1999), p. 55. DOI: 10.1609/aimag.v20i1.1440. URL: https://ojs.aaai.org/index.php/aimagazine/article/view/1440.

[17] Roger Almeida Leite et al. "Visual Analytics for Fraud Detection: Focusing on Profile Analysis". In: *Proceedings of the Eurographics / IEEE VGTC Conference on Visualization: Posters.* EuroVis '16. Groningen, The Netherlands: Eurographics Association, 2016, pp. 45–47.

[18] Chika Sakoda et al. "Visualization for assisting rule definition tasks of credit card fraud detection systems". In: *IIEEJ Image Electronics and Visual Computing Workshop.* 2010.

[19] W. Didimo et al. "An advanced network visualization system for financial crime detection". In: *2011 IEEE Pacific Visualization Symposium.* 2011, pp. 203–210. DOI: 10.1109/PACIFICVIS.2011.5742391.

[20] Walter Didimo, Giuseppe Liotta, and Fabrizio Montecchiani. "Vis4AUI: Visual Analysis of Banking Activity Networks." In: *GRAPP/IVAPP.* 2012, pp. 799–802.

[21] R. Chang et al. "WireVis: Visualization of Categorical, Time-Varying Data From Financial Transactions". In: *2007 IEEE Symposium on Visual Analytics Science and Technology.* 2007, pp. 155–162.

[22] T. Kohonen. "The self-organizing map". In: *Proceedings of the IEEE* 78.9 (1990), pp. 1464–1480. DOI: 10.1109/5.58325.

[23] Nicoleta Rogovschi, Mustapha Lebbah, and Younès Bennani. "A self-organizing map for mixed continuous and categorical data". In: *Int. Journal of Computing* 10.1 (2011), pp. 24–32.

[24] Chung-Chian Hsu and Shu-Han Lin. "Visualized analysis of mixed numeric and categorical data via extended self-organizing map." eng. In: *IEEE Trans Neural Netw Learn Syst* 23.1 (Jan. 2012), pp. 72–86. ISSN: 2162-237X (Print); 2162-237X (Linking). DOI: 10.1109/TNNLS.2011.2178323.

[25] C. Hsu and C. Kung. "Incorporating unsupervised learning with self-organizing map for visualizing mixed data". In: *2013 Ninth International Conference on Natural Computation (ICNC).* 2013, pp. 146–151. DOI: 10.1109/ICNC.2013.6817960.

[26] Wei-Shen Tai and Chung-Chian Hsu. "Growing Self-Organizing Map with cross insert for mixed-type data clustering". In: *Applied Soft Computing* 12.9 (2012), pp. 2856–2866. ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2012.04.004. URL: http://www.sciencedirect.com/science/article/pii/S1568494612001731.

[27] Chung-Chian Hsu. “Generalizing self-organizing map for categorical data”. In: *IEEE Transactions on Neural Networks* 17.2 (2006), pp. 294–304. DOI: 10.1109/TNN.2005.863415.

[28] Carmelo del Coso et al. “Mixing numerical and categorical data in a Self-Organizing Map by means of frequency neurons”. In: *Applied Soft Computing* 36 (2015), pp. 246–254. ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2015.06.058. URL: http://www.sciencedirect.com/science/article/pii/S1568494615004512.

[29] EL Koua. “Using self-organizing maps for information visualization and knowledge discovery in complex geospatial datasets”. In: *Proceedings of 21st int. cartographic renaissance (ICC)* (2003), pp. 1694–1702.

[30] Zeqian Shen et al. “BiblioViz: a system for visualizing bibliography information”. In: *Proceedings of the 2006 Asia-Pacific Symposium on Information Visualisation-Volume 60*. Australian Computer Society, Inc. 2006, pp. 93–102.

[31] Dominik Olszewski. “Fraud detection using self-organizing map visualizing the user profiles”. In: *Knowledge-Based Systems* 70 (2014), pp. 324–334. ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2014.07.008.

[32] Matija Milosevic et al. “Visualization of trunk muscle synergies during sitting perturbations using self-organizing maps (SOM).” eng. In: *IEEE Trans Biomed Eng* 59.9 (Sept. 2012), pp. 2516–2523. ISSN: 1558-2531 (Electronic); 0018-9294 (Linking). DOI: 10.1109/TBME.2012.2205577.

[33] César A. Astudillo and B. John Oommen. “Topology-oriented self-organizing maps: a survey”. In: *Pattern Analysis and Applications* 17.2 (2014), pp. 223–248. DOI: 10.1007/s10044-014-0367-9. URL: https://doi.org/10.1007/s10044-014-0367-9.

[34] Jorge M. L. Gorricha and Victor J. A. S. Lobo. “On the Use of Three-Dimensional Self-Organizing Maps for Visualizing Clusters in Georeferenced Data”. In: *Information Fusion and Geographic Information Systems: Towards the Digital Ocean*. Ed. by Vasily V. Popovich et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 61–75. ISBN: 978-3-642-19766-6. DOI: 10.1007/978-3-642-19766-6_6.

[35] Alessandra Marli M. Morais, Marcos Gonçalves Quiles, and Rafael D. C. Santos. “Icon and Geometric Data Visualization with a Self-Organizing Map Grid”. In: *Computational Science and Its Applications – ICCSA 2014*. Ed. by Beniamino Murgante et al. Cham: Springer International Publishing, 2014, pp. 562–575. ISBN: 978-3-319-09153-2.

[36] Gennady Andrienko et al. “A framework for using self-organising maps to analyse spatio-temporal patterns, exemplified by analysis of mobile phone usage”. In: *Journal of Location Based Services* 4.3-4 (2010), pp. 200–221. DOI: 10.1080/17489725.2010.532816.

[37] Barbara Furletti et al. “Identifying Users Profiles from Mobile Calls Habits”. In: *Proceedings of the ACM SIGKDD International Workshop on Urban Computing*. UrbComp ’12. Beijing, China: Association for Computing Ma-

chinery, 2012, pp. 17–24. ISBN: 9781450315425. DOI: 10.1145/2346496.2346500.

[38] T. Schreck et al. “Visual cluster analysis of trajectory data with interactive Kohonen Maps”. In: *2008 IEEE Symposium on Visual Analytics Science and Technology.* 2008, pp. 3–10. DOI: 10.1109/VAST.2008.4677350.

[39] Y. Kameoka et al. “Customer segmentation and visualization by combination of self-organizing map and cluster analysis”. In: *2015 13th International Conference on ICT and Knowledge Engineering (ICT Knowledge Engineering 2015).* 2015, pp. 19–23. DOI: 10.1109/ICTKE.2015.7368465.

[40] Ron Wehrens and Lutgarde M. C. Buydens. “Self- and Super-organizing Maps in R: The kohonen Package”. In: *Journal of Statistical Software, Articles* 21.5 (2007), pp. 1–19. ISSN: 1548-7660. DOI: 10.18637/jss.v021.i05. URL: https://www.jstatsoft.org/v021/i05.

[41] Tobias Schreck et al. “Trajectory-Based Visual Analysis of Large Financial Time Series Data”. In: *SIGKDD Explor. Newsl.* 9.2 (Dec. 2007), pp. 30–37. ISSN: 1931-0145. DOI: 10.1145/1345448.1345454. URL: https://doi.org/10.1145/1345448.1345454.

[42] Peter Sarlin and Tomas Eklund. “Fuzzy Clustering of the Self-Organizing Map: Some Applications on Financial Time Series”. In: *Advances in Self-Organizing Maps.* Ed. by Jorma Laaksonen and Timo Honkela. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 40–50. ISBN: 978-3-642-21566-7.

[43] P. Sarlin. “Sovereign debt monitor: A visual Self-organizing maps approach”. In: *2011 IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CIFEr).* 2011, pp. 1–8. DOI: 10.1109/CIFER.2011.5953556.

[44] Krešimir Šimunić. “Visualization of Stock Market Charts”. In: *In Proceedings from the 11th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, Plzen-Bory (CZ).* 2003.

[45] Jock Mackinlay. “Automating the Design of Graphical Presentations of Relational Information”. In: *ACM Trans. Graph.* 5.2 (Apr. 1986), pp. 110–141. ISSN: 0730-0301. DOI: 10.1145/22949.22950. URL: https://doi.org/10.1145/22949.22950.

[46] Remco Chang et al. “Scalable and Interactive Visual Analysis of Financial Wire Transactions for Fraud Detection”. In: *Information Visualization* 7.1 (2008), pp. 63–76. DOI: 10.1057/palgrave.ivs.9500172.

[47] Jens Olsson and Martin Boldt. “Computer forensic timeline visualization tool”. In: *Digital Investigation* 6 (2009). The Proceedings of the Ninth Annual DFRWS Conference, S78–S87. ISSN: 1742-2876. DOI: https://doi.org/10.1016/j.diin.2009.06.008. URL: http://www.sciencedirect.com/science/article/pii/S1742287609000425.

[48] Mathieu Jacomy et al. “ForceAtlas2, a Continuous Graph Layout Algorithm for Handy Network Visualization Designed for the Gephi Software”.

In: *PLOS ONE* 9.6 (June 2014), pp. 1–12. DOI: 10.1371/journal.pone.0098679. URL: https://doi.org/10.1371/journal.pone.0098679.