VaBank: Visual Analytics for Banking Transactions

Catarina Maçãs, Evgheni Polisciuc, and Penousal Machado
University of Coimbra, Centre for Informatics and Systems of the University of Coimbra
Department of Informatics Engineering
Coimbra, Portugal
{*cmacas,evgheni,machado*}*@dei.uc.pt*

***Abstract*—To analyse and detect fraudulent patterns in banking transactions, most fraud analysts use spreadsheets which makes the overall process time-consuming and complex. In this article, we propose a visualization tool that aims to ease the analysis of banking transactions over time and the detection of the transactions' topology and of suspicious behaviours. Our main contributions are: (i) a user-centred visual tool, developed with the aid of fraud experts; (ii) a method that characterises the transactions topology through a self-organising algorithm; (iii) the visual characterisation of transactions through complex glyphs; and (iv) a user study to assess the tool effectiveness.**

***Keywords*-visual analytics, self-organising maps, glyph, profiling, fraud, banking transactions**

I. Introduction

The analysis of financial transactions can be an overwhelming task for bank and fraud analysts. However, such analysis is important for the detection of suspicious behaviours that may lead to the finding of fraudulent activities. Based on our interaction with fraud analysts, we could summarise their main analysis goals: to understand the temporal evolution of a set of transactions—usually grouped by an attribute, such as client ID—and to search for patterns and common characteristics among transactions. Usually, they analyse the data through spreadsheets, which are complex, time-consuming, and may be inappropriate for complex datasets [1].

Through the combination of computational strategies and our visual cognitive intelligence [2], visual analytics can facilitate the analysis of transactional data and enhance the representation of transactions over time. In addition, the application of clustering algorithms and the visualization of their results can be a reliable approach for the analysis of patterns in transactions. For example, self-organising maps (SOMs) have already proven its usefulness and robustness for the analysis of large amounts of data [3]. Also, with the visualization of its results, it is possible to provide a visual summary of the data topology and ease the interpretation of behaviours in a single image [4, 5].

This project departed from a collaboration with Feedzai, a world leading fraud detection company. They gave us access to a properly anonymised dataset of banking data to develop a visual analytics tool that enables their analysts to: (i) inspect collections of transactions in a single place; (ii) understand the overall behaviours of a set of transactions; and (iii) detect the most common types of transactions. Our tool is divided into two main views—the visualization of the transactions history and the visualization of the transactions topology. The latter relies on a SOM algorithm and the visualization of its results through two visualization techniques: matrix and force-directed projections. Both aim to represent a group of transactions and enable the understanding of the transaction's characteristics and the most common ones. In the transaction history visualization, we provide a set of analytical features that enable the user to navigate, explore, and analyse the transactions over time. To evaluate our tool and verify its usefulness in the analysis of bank data, we conducted a user study, held with fraud analysts. The results showed that with our tool the analysts could easily analyse the transactions, detect suspicious behaviours, and understand the transactions topology. The analysts also referred that our tool could substantially improve their line of work.

II. background

A. Visualisation in Finance domain

The majority of fraud prevention companies employ Machine Learning (ML) to detect patterns of fraud [6, 7] and discard transactions which are promptly classified as fraud by their systems. However, as the mechanisms of fraud are always evolving and adapting, fraud prevention companies also employ fraud analysts to manually analyse suspicious activities and validate the system's classifications. Currently, to evaluate transactions, most analysts use spreadsheets and tabular forms which support various operations to extract more detailed information. Nevertheless, they are not effective at providing a clear representation of patterns, trends, and correlations hidden in data [1]. For this reason, fraud analysts recognise the relevance of Visual Analytics, as it enables them to better understand the data and draw conclusions more rapidly [8].

From the state of the art, some visualizations have been presented and discussed in the financial domain [9–11]. Nonetheless, we found only one related to the visualization of bank data. Wire Viz is a coordinated visualization tool that aims to identify specific keywords within a set of transactions. They use different views to depict relationships among keywords and accounts over time. Their goal is

978-1-7281-9134-8/20/$31.00 ©2020 IEEE
DOI 10.1109/IV51561.2020.00062

to give an overview of the data, provide the ability to aggregate and organise groups of transactions and compare individual records [1]. However, this project uses a different type of dataset, containing transactions to and from other banks, whereas in our project, we only have access to the transactions made in one bank. With this, we are not able to follow the connections between different transactions.

B. Self-organising Maps

A self-organising map (SOM) is a method for dimensionality reduction that preserves topological and metric relationships of the input data. As such, SOMs are a powerful tool for communicating complex, nonlinear relationships among high-dimensional data through simple graphical representations.

In the present work, we focus on SOMs which work with mixed data, i.e., data comprised by numerical and categorical attributes. The topological self-organising algorithm for analysing mixed variables was proposed in [12], in which categorical data is encoded to binary variables. Later, other works appeared in which categorical values are dealt through semantic similarities [13–15], distance hierarchies, and frequency-based distances [16].

1) SOM Visualisation: The visualization of SOMs is typically concerned with the projection of neurons into a 2D 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 grey scale colour palette. This visual mapping can be used in the detection of clusters [17, 18] or in the definition of thresholds [19]. Additionally, hexagonal grids [20] can also be used [21]. To improve the reading and understanding of each neuron, some works used complex glyphs, such as line graphs—to represent, for example, the evolution of call logs [22] or trajectories [23]—and radial graphs—to represent, for example, the consumption values [24, 25] or the weights of each feature of the SOM [26].

2) SOM in Finance: The application of SOM algorithms to analyse transactional data have been applied in a variety of works. The majority of the found works apply SOMs to provide an analytical view on the financial market trajectories [23, 27, 28], to analyse their stability, and to monitor multidimensional financial data [29]. Others apply SOMs to better comprehend the stock market dynamics [30] or to analyse the financial performance of companies [3].

III. REQUIREMENTS AND TASKS

From our collaboration with Feedzai, a fraud detection company, we could held several meetings with analysts and better define the domain and requirements for the analysis of bank data. The analysts emphasised two main tasks: **[T1]** the comprehension of the transactions history; and **[T2]** the detection of the most common types of transactions. The latter, is specially important as it enables the distinction between typical and atypical behaviours. Then, the analysts described their line of work, and summarised five requirements:

[R1] Search by field. The analysts usually sort the data by a certain field (e.g., client ID, Country of IP) and select all transactions with the same value. The implementation of a mechanism that eases this task is of utmost importance to speed up the analysis process and facilitate the grouping of different transactions;

[R2] Distinguish amount values. When dealing with transactions, the amount value can reveal fraudulent actions, being values above a certain threshold worth of more detailed analysis. Hence, the visual sorting of transactions by amount can enhance the detection of suspicious transactions;

[R3] Distinguish transactions. By visually characterising each transaction, the analysts can focus their attention on transactions of the same type and perceive the evolution of their amounts along time, detecting atypical behaviours;

[R4] Search common fields. When dealing with spreadsheets, the analysts have difficulties in detecting transactions that share common attributes. This if of utmost importance when analysing fraudulent transactions which can share attributes with others. Hence, it is important to enable the highlight of transactions with similar attributes;

[R5] Detect typical transactions. Detect the most common types of transactions can aid the analyst in the perception of unusual transactions, possibly related to fraud. Hence, it is important to characterise typical transactions.

IV. BANK DATA

We worked with an anonymised dataset which contains only transactions made by the clients of a bank—there is no data on the transactions that each client receives. Each transaction of the dataset is characterised by attributes corresponding to the: client (e.g., ID, IBAN), location (e.g., client IP, Country IP), amount (e.g., amount, currency), transaction type (e.g., transaction 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. Any client can have transactions of both types. The transactions also have a descriptor, composed by two or three acronyms, that characterise the transaction according to: the interface used; the type of operation (e.g., national, international, loan); and whether it is for a new beneficiary or not. These acronyms must be known by the analysts so they can properly analyse the transactions. This task has a high level of difficulty as these descriptors can have different combinations (**[R3]**). Additionally, all transactions are labelled by the bank as fraud or not.

A. SOM Algorithm

We applied a variant of the SOM algorithm prepared to handle mixed data—Frequency neuron Mixed Self-Organising Map (FMSOM) [16]. It consists on 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. Also, the FMSOM model allowed us to adapt it to define the dissimilarity between neurons, used in the visualization of the transaction's topology.

1) 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 IV and cannot be fully revealed due to the specificity and sensibility 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* since the last transaction and until the next transaction are previously calculated and are intended to capture the patterns of the transactional regularity.

2) Dissimilarity Metric: We applied different measures to compute the distances between neurons—traditional Euclidean distance for continuous values, and the measure based on probabilities (described in [16]) 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 [16], the dissimilarity measure between neuron and the input feature vector consist on 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 [16].

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 $d(W_i, W_j) = Dn(W_i, W_j) + Dk(W_i, W_j)$

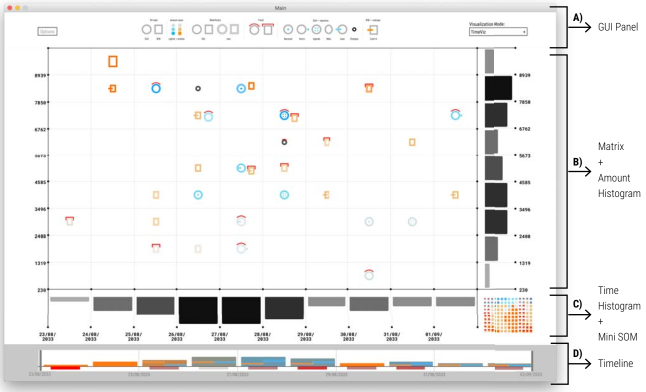

Figure 1. Transaction History view and its components.

V. VABANK TOOL

VaBank aims to answer to the tasks referred in III and is divided into three views: the transaction history **[T1]**; the transactions topology **[T2]**; and the transactions relations **[T2]**. The first arranges all transactions by time and amount. The last two display the results of the SOM algorithm (see IV-A) in a grid and through a force-directed graph, respectively. A video showing the interaction with the tool can be seen in: https://vimeo.com/444222426.

All views have access to a GUI panel (Fig. 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 sorted in an ascending way, according to the number of transactions of each client, and the user can scroll and select different clients. In this panel, the user can also access a list of other fields and select one to group the transactions **[R1]**. On the upper right corner of the GUI panel, a dropdown menu enables the user to change the visualization view. In the middle of the GUI panel, a caption is shown to aid in the analysis of the transaction representation (Fig. 1, A).

A. Representation of a transaction

All views share one visual element: the transactions. To ease their distinction and visual characterisation, we implemented a glyph **[R3]**. The glyphs are composed by three levels of visual detail. These levels were defined together with the company's analysts, according to the types of information more relevant when analysing bank data. First, the analysts aim to distinguish online from business transactions. Then, it is necessary to analyse the transaction amount and whether it was considered as fraud or not. These three characteristics represent the first level of visual impact. Then, the analysts want to drill down and distinguish between: inbound and

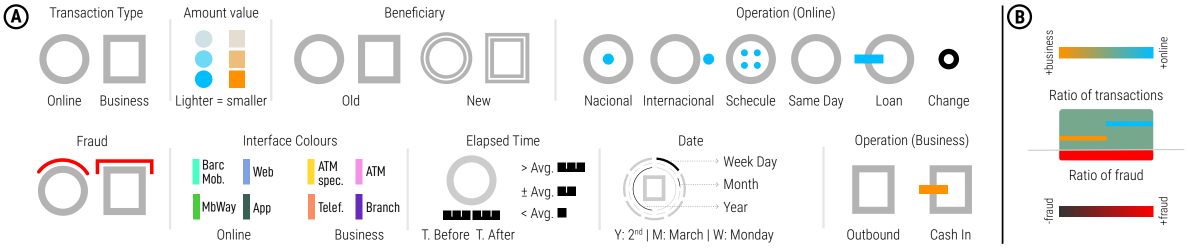

Figure 2. Glyph elements that characterise each transaction (side A) and timeline bar composition and respective colour ranges (side B)

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 are defined as less important than the described above. For this reason, they are grouped into the third level of visual impact, having a lower visual impact.

As colour has a high impact on visualization [31], 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. Then we use saturation to represent the amount, the brighter the colour, the higher the amount. As small differences in saturation would be imperceptible, we defined three levels of saturation to represent: 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. We also use shapes to emphasise the distinction between transaction types—a circle for online transactions, and a rectangle for business. To represent fraud, we place a red line above the main shape (see Fig. 2, A).

The transactions' shape are 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. The inbound is represented by the same symbol in online (i.e., Loan) and business (i.e., Cash In) transactions: a centred horizontal rectangle positioned on the left. The outbound operations are represented as depicted in Fig. 2. 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 (Fig. 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 circle with different radius centred in the main shape, being the year the smallest, and the day of the week the biggest (Fig. 2). To distinguish periods of time, we divide the stroke in: 7 wedges, for the days of the week; 12 wedges for the months; and, for the years, in the total number of years of the data. All wedges are coloured in light grey, except the wedge that marks the transaction date, coloured in black. The day of the week has a thicker wedge, as the analysts referred to it as 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 time distances. As with the amount thresholds, we defined three levels of time distances, computed in the same way. These three levels are represented as depicted in Fig. 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 interface colour (Fig. 2, A).

The glyphs used in the views concerning the SOM results 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 represented.

B. *Transaction History*

In this view, we implement a set of visualization models to display different data aggregations. The main representation, which occupies more canvas space, is the the transaction matrix (Fig. 1, B). It divides the space in different ranges of amounts on the y-axis and temporal values on the x-axis **[R2]**. The transactions 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 V-A) occur within the same cell, they are aggregated and its glyph grows in size. The placement within each cell is made through a circle packing algorithm which starts by placing the biggest glyph in the middle of the cell and the others around it. The user can hover each glyph and see more details—Country IP, amount, beneficiary, and number of transactions. If the user clicks on a glyph, these details are fixed in the canvas. Then, the user can click on an attribute to highlight all transactions that share that attribute **[R4]**.

In the bottom and right sides of the matrix, histograms are drawn to show the total number of transactions per column and row, respectively (Fig. 1, B and C). The histogram's bars are coloured according to the number of transactions: the darker, the higher the number of transactions. By hovering each bar, the total number of transactions is shown. In the bottom right corner of the matrix area, we draw a small matrix of glyphs that represents the result of a SOM

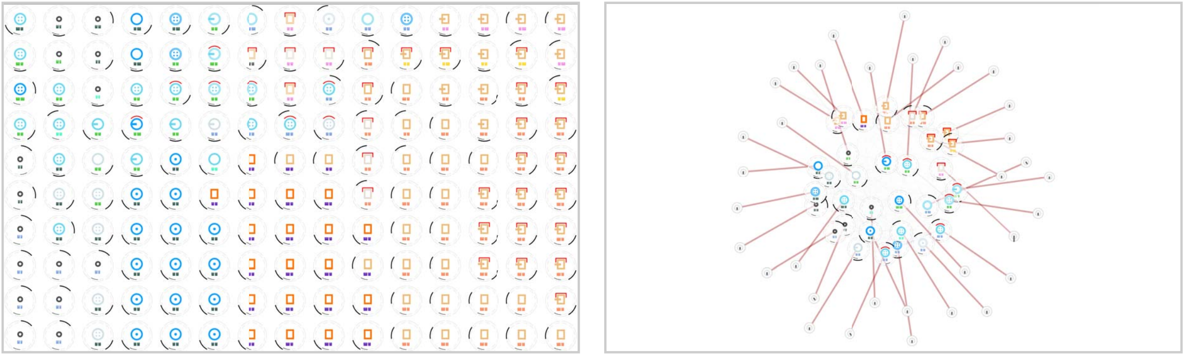

Figure 3. Projections of the SOM results for the same bank client through the matrix projection (left) and force-directed graph (right).

algorithm, concerning: amount, transaction type, and fraud (Fig. 1, C). With this, we aim to enhance the understanding of typical/atypical transactions **[R5]**.

In the bottom, we placed an interactive timeline, so the user can select and visualise different periods of time in the data (Fig. 1, D). The different time periods are defined by a hierarchical aggregation algorithm (see Section V-B1 for more details). The timeline is divided vertically in two parts. In the upper part, we represent the number of transactions with bars. Each bar is drawn as follows: (i) its height represents the total number of transactions; (ii) the main colour of the bar is defined by a gradient between blue and orange—the more blue, the higher the number of online transactions, the more orange, the higher the number of business transactions (Fig. 2, B). With this, we aim to overview which type of transaction occurs the most. To give more details, we represent the quantity of each transaction type with a bar placed vertically in the main bar, according to the percentage of occurrence, and coloured, according to the transaction type. In the bottom part of the timeline, we place a rectangle with a predefined height if one or more fraudulent transactions occur. This rectangle 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 more red the bar will be (Fig. 2, B). If no fraud occurs, no bar is drawn. The user can hover each bar to see a set of statistics (e.g., percentages of online and business transactions, and percentage of fraud).

1) Hierarchical Temporal Aggregation: Our adaptive timeline algorithm takes as arguments the available space to draw the timeline and the minimal width of a time bar (which represents a range of time). The main goal is to allow the selection of any subset of the data and enable the timeline to adapt its granularity and adjust the size of time bars, solving the problem of fixed timelines. The main issue of these timelines is that selecting a subset of data can lead to cluttered timelines, uneven distribution of the time bars, or the inefficient use of space, due to the time granularity.

Our algorithm follows an iterative top-down approach. We start at the biggest time unit existing in the computation systems (e.g., epoch), and descend, 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 time bars. The algorithm have to meet a single criteria 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.

C. *Transactions Topology*

We applied a SOM algorithm to enhance the detection of atypical transactions, which can be related to fraudulent behaviours (Fig. 3). The results of the SOM are visualised through a matrix or a force-directed projections. The goal of the first is to represent the distribution of different types of transactions present in the data and extrapolate at a higher level the characteristics of the dataset. The goal of the second is to express the relations among clustered data and emphasise the most typical transaction, allowing a more detailed analysis of the dataset.

1) Transactions Matrix: In this projection, 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. This approach enables the analyst to analyse the most common types of transactions. However, it 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 more representative of the dataset. The latter task is specially difficult to achieve when more than one feature is being represented in the glyphs. To accomplish them, we implemented a second approach, in which we place each neuron within a force-directed graph and represent their relations to the transactions. We aim to achieve a better comprehension of the transactions profile.

2) Transactions Relations: For the force-directed graph, neurons and 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 (Fig. 3, right). Our implementation of the graph is based on the Force Atlas 2 algorithm [32]. All the nodes have forces of repulsion from each other so they do not overlap. However, only nodes which dissimilarity is below a predefined threshold have forces of attraction, creating visual clusters defined by the SOM topology. We added a gravitational force, attracting all nodes to the centre of the canvas. The higher the number of connections between nodes, the higher this gravitational force. With this, clusters 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 a best matching unit (BMU) in the training process of the SOM are represented, leading to a more representative graph. Also, the transactions which have the same neuron as BMU are aggregated and defined as a node. Their forces of attraction are defined by the average force of attraction to other neurons.

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

We use a line 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 nodes proximity, but as they should have less visual emphasis, their opacity and thickness diminishes according to the similarity.

VI. EVALUATION

To evaluate the tool usefulness and effectiveness in the analysis of banking transactions, we performed a user testing with 5 fraud analysts—which were not present during the tool development. Their aim was to perform a set of tasks and to analyse two clients' transactions through the interaction with the tool. The tests were performed as follows: (i) we introduced the transactions representation, the views of the tool and its interaction mechanisms; (ii) we asked the analysts to perform 18 tasks concerning: transaction history (6); interpretability of the glyphs (4); SOM matrix (4); and SOM graph (4); (iii) the analysts analysed two clients in terms of behaviour and fraud; and (iv) the analyst gave 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, in the end of each, the analysts were asked to rate the difficulty of each exercise and certainty of their answer—from 1 to 5.

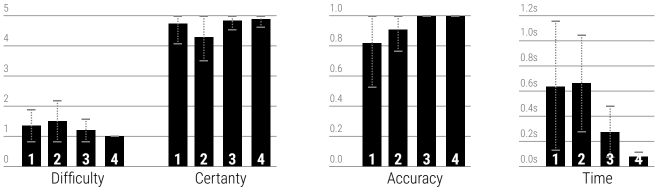

Figure 4. Difficulty, Certainty, Accuracy and Time for the 4 tasks groups.

The tasks were divided into 4 groups, depending on the component they aim to validate: **G1** Transaction History model; **G2** Transaction glyphs; **G3** SOM Matrix; and **G4** SOM Graph. In the transaction history view, we tested the analysts ability to comprehend temporal patterns, and the transactions distribution concerning time and amount values. In the SOM projections, we aimed to compare both views and perceive which 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 projections. Additionally, the third part of the tests aims to understand the usefulness of the tool and its ability to aid the analysts to detect suspicious patterns and possible frauds.

A. Results and Discussion

Fig. 4 summarises the results concerning difficulty, certainty, accuracy, and duration for each group of tasks. Hereafter, we further analyse each group and discuss the results from the third part of the test and the analysts feedback.

1) Transaction History and Glyphs: Although all difficulty ratings are low, the tasks related to the analysis of the Transaction History view (**G1**) and the glyphs (**G2**) raised more difficulty in comparison to the others. Regarding the Transaction History, the analysts had some difficulties in interpreting the positioning of the glyphs in the grid and the histograms. Some analysts, for the task of "In which period of time the business attribute had the highest amount?", 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 about the tool. The analysts also had some difficulty in interpreting the glyphs, which made their certainty to be slightly lower than the other groups. However, the accuracy of this task is higher than the Transaction History tasks. We could perceive that, as the glyphs were complex, they were not certain if they were characterising all their attributes correctly, which caused lower rates of certainty.

2) SOM *Visualisation:* The groups of tasks related to the SOM analysis took less time to perform (20 seconds, on average), had 100% of accuracy, and were the ones in which the analysts had more certainty of their answers and less difficulty in completing the tasks. Comparing both projections, the graph (**G4**) tasks were performed more quickly and with less difficulty. This can be explained by the fact that, as the graph is less complex (and has less glyphs), for the same tasks the analysts could analyse more quickly the glyphs and their relations.

3) VaBank analysis: The second part of the tests was concerned with the free exploration and analysis of the transactions of two clients. These clients have two different behaviours: client 1 has a suspicious behaviour in the end of his/her transactional history, and client 2 commits fraud in the beginning of his/her transactional history. The majority of the analysts identified client 1 as a suspicious case, in which he starts with periodic transactions and, in the end, start to do transactions with higher amounts and at a higher rate. Client 2 was instantly classified has fraudulent, for his attempts of doing several transactions with high amounts for different accounts. Also, most analysts referred to client 2 has an hacked account. All analysts could interact properly with the tool. They stated that after the tasks completion they were more familiarised with the tool, and could use easily all functionalities.

4) User Feedback: In the end of each test, the analysts rated each view in terms of aesthetics, interpretability, analytical usefulness, and learning curve. The Transaction History view, got the higher rate in terms of aesthetics and aid. Additionally, it was defined as the easier to learn and interpret. This enhance the fact that, although in the tasks it was considered challenging, after interaction it got easier to interpret. This view was well received by the analysts which defined it as a good auxiliary for their work. Concerning the graph and matrix views of the SOM, with the matrix view the analysts took more time to complete the tasks and rated it with higher values of difficulty. However, the matrix grid was seen as a better aid to analyse the transaction patterns and was also defined has easier to learn.

VII. CONCLUSION

We presented a user-centred visualization tool for the analysis of banking transactions data. Through the collaboration with Feedzai, one of the main fraud prevention companies, we were able to define the tasks for the analysis of such cases. This led us to the definition of the main design requirements for the implementation of a visualization tool focused 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 SOM algorithm; and (iv) the projection of the SOM results into a matrix and a force-directed graph. We validated the visualizations through formative and summative evaluations with experts in fraud detection. The results showed that the tool was well received by the analysts as it could enhance their analysis of transactional data.

ACKNOWLEDGMENT

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 FCT, under the grant SFRH/BD/129481/2017.

REFERENCES

[1] R. Chang, M. Ghoniem, R. Kosara, W. Ribarsky, J. Yang, E. Suma, C. Ziemkiewicz, D. Kern, and A. Sudjianto, "Wirevis: Visualization of categorical, time-varying data from financial transactions," in *2007 IEEE Symposium on Visual Analytics Science and Technology*. IEEE, 2007, pp. 155–162.

[2] V. L. Lemieux, B. W. Shieh, D. Lau, S. H. Jun, T. Dang, J. Chu, and G. Tam, "Using visual analytics to enhance data exploration and knowledge discovery in financial systemic risk analysis: The multivariate density estimator," *iConference 2014 Proceedings*, 2014.

[3] T. Eklund, B. Back, H. Vanharanta, and A. Visa, "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*, S. Wrycza, Ed., vol. 1. AIS, 2002.

[4] M. Y. Kiang and A. Kumar, "An evaluation of self-organizing map networks as a robust alternative to factor analysis in data mining applications," *Information Systems Research*, vol. 12, no. 2, pp. 177–194, 2001.

[5] A. Costea, A. Kloptchenko, B. Back, I. Ivan, and I. Rosca, "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.

[6] R. J. Bolton and D. J. Hand, "Statistical fraud detection: A review," *Statistical science*, pp. 235–249, 2002.

[7] E. W. Ngai, Y. Hu, Y. Wong, Y. Chen, and X. Sun, "The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature," *Decision support systems*, vol. 50, no. 3, pp. 559–569, 2011.

[8] W. N. Dilla and R. L. Raschke, "Data visualization for fraud detection: Practice implications and a call for

future research," *International Journal of Accounting Information Systems*, vol. 16, pp. 1–22, 2015.

[9] R. A. Leite, T. Gschwandtner, S. Miksch, E. Gstrein, and J. Kuntner, "Visual analytics for event detection: Focusing on fraud," *Visual Informatics*, vol. 2, no. 4, pp. 198–212, 2018.

[10] S. Ko, I. Cho, S. Afzal, C. Yau, J. Chae, A. Malik, K. Beck, Y. Jang, W. Ribarsky, and D. S. Ebert, "A survey on visual analysis approaches for financial data," in *Computer Graphics Forum*, vol. 35. Wiley Online Library, 2016, pp. 599–617.

[11] M. Dumas, M. J. McGuffin, and V. L. Lemieux, "Financevis. net-a visual survey of financial data visualizations," in *Poster Abstracts of IEEE Conference on Visualization*, vol. 2, 2014.

[12] N. Rogovschi, M. Lebbah, and Y. Bennani, "A self-organizing map for mixed continuous and categorical data," *Int. Journal of Computing*, vol. 10, no. 1, pp. 24–32, 2011.

[13] C.-C. Hsu and S.-H. Lin, "Visualized analysis of mixed numeric and categorical data via extended self-organizing map," *IEEE transactions on neural networks and learning systems*, vol. 23, no. 1, 2011.

[14] C.-C. Hsu and C.-H. Kung, "Incorporating unsupervised learning with self-organizing map for visualizing mixed data," in *2013 Ninth International Conference on Natural Computation (ICNC)*. IEEE, 2013.

[15] W.-S. Tai and C.-C. Hsu, "Growing self-organizing map with cross insert for mixed-type data clustering," *Applied Soft Computing*, vol. 12, no. 9, 2012.

[16] C. Del Coso, D. Fustes, C. Dafonte, F. J. Nóvoa, J. M. Rodríguez-Pedreira, and B. Arcay, "Mixing numerical and categorical data in a self-organizing map by means of frequency neurons," *Applied Soft Computing*, vol. 36, pp. 246–254, 2015.

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

[18] Z. Shen, M. Ogawa, S. T. Teoh, and K.-L. Ma, "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.

[19] D. Olszewski, "Fraud detection using self-organizing map visualizing the user profiles," *Knowledge-Based Systems*, vol. 70, pp. 324–334, 2014.

[20] M. Milosevic, K. M. V. McConville, E. Sejdic, K. Masani, M. J. Kyan, and M. R. Popovic, "Visualization of trunk muscle synergies during sitting perturbations using self-organizing maps (som)," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 9, pp. 2516–2523, 2012.

[21] C. A. Astudillo and B. J. Oommen, "Topology-oriented self-organizing maps: a survey," *Pattern analysis and applications*, vol. 17, no. 2, pp. 223–248, 2014.

[22] B. Furletti, L. Gabrielli, C. Renso, and S. Rinzivillo, "Identifying users profiles from mobile calls habits," in *Proceedings of the ACM SIGKDD int. workshop on urban computing*. ACM, 2012, pp. 17–24.

[23] T. Schreck, J. Bernard, T. Von Landesberger, and J. Kohlhammer, "Visual cluster analysis of trajectory data with interactive kohonen maps," *Information Visualization*, vol. 8, no. 1, pp. 14–29, 2009.

[24] Y. Kameoka, K. Yagi, S. Munakata, and Y. Yamamoto, "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)*. IEEE, 2015, pp. 19–23.

[25] C. Maçãs, E. Polisciuc, and P. Machado, "Glyphsome: Using SOM with data glyphs for customer profiling," in *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISIGRAPP 2020, Volume 3: IVAPP, Valletta, Malta, February 27-29, 2020*, 2020.

[26] R. Wehrens, L. M. Buydens *et al.*, "Self-and super-organizing maps in r: the kohonen package," *Journal of Statistical Software*, vol. 21, no. 5, pp. 1–19, 2007.

[27] T. Schreck, T. Tekušová, J. Kohlhammer, and D. Fellner, "Trajectory-based visual analysis of large financial time series data," *SIGKDD Explor. Newsl.*, vol. 9, no. 2, p. 30–37, Dec. 2007.

[28] P. Sarlin and T. Eklund, "Fuzzy clustering of the self-organizing map: some applications on financial time series," in *International Workshop on Self-Organizing Maps*. Springer, 2011, pp. 40–50.

[29] P. Sarlin, "Sovereign debt monitor: A visual self-organizing maps approach," in *2011 IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CIFEr)*. IEEE, 2011, pp. 1–8.

[30] K. 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.

[31] J. Mackinlay, "Automating the design of graphical presentations of relational information," *Acm Transactions On Graphics (Tog)*, vol. 5, no. 2, pp. 110–141, 1986.

[32] M. Jacomy, T. Venturini, S. Heymann, and M. Bastian, "Forceatlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software," *PloS one*, vol. 9, no. 6, 2014.