


STALITA: Innovative Platform for Bank Transactions Analysis

David Jesenko ^{1,*} , Štefan Kohek ¹ , Borut Žalik ¹ , Matej Brumen ¹, Domen Kavran ¹, Niko Lukač ¹ , Andrej Živec ² and Aleksander Pur ²

¹ Faculty of Electrical Engineering and Computer Science, University of Maribor, Koroška cesta 46, SI-2000 Maribor, Slovenia

² Ministry of the Interior, Štefanova ulica 2, SI-1000 Ljubljana, Slovenia

* Correspondence: david.jesenko@um.si; Tel.: +386-2-220-7476

Abstract: Acts of fraud have become much more prevalent in the financial industry with the rise of technology and the continued economic growth in modern society. Fraudsters are evolving their approaches continuously to exploit the vulnerabilities of the current prevention measures in place, many of whom are targeting the financial sector. To overcome and investigate financial frauds, this paper presents STALITA, which is an innovative platform for the analysis of bank transactions. STALITA enables graph-based data analysis using a powerful Neo4j graph database and the Cypher query language. Additionally, a diversity of other supporting tools, such as support for heterogeneous data sources, force-based graph visualisation, pivot tables, and time charts, enable in-depth investigation of the available data. In the Results section, we present the usability of the platform through real-world case scenarios.

Keywords: Neo4j; platform; bank transactions; graph analysis; graph visualisation; fraud; investigation



Citation: Jesenko, D.; Kohek, Š.; Žalik, B.; Brumen, M.; Kavran, D.; Lukač, N.; Živec, A.; Pur, A. STALITA: Innovative Platform for Bank Transactions Analysis. *Appl. Sci.* **2022**, *12*, 12492. <https://doi.org/10.3390/app122312492>

Academic Editor: Yoshiyasu Takefuji

Received: 24 October 2022

Accepted: 4 December 2022

Published: 6 December 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Nowadays, cyber security and digital forensics are the foundations for ensuring security for all digital flows of information. Both main disciplines can be divided into smaller, more specific fields. The discipline of digital forensics encompasses many different areas, including computer forensics, network forensics, mobile forensics, IoT forensics, and malware forensics [1]. The emerging sub-discipline of digital forensics also covers financial technologies, which include different money activities such as payments, fund transfers, and other financial transactions. As they are intended for a wide range of users, they include those with good and bad intentions. The latter are leveraging financial transactions for fraud, extortion, money laundering, and other financial activities in the criminal underground. Those incidents may use and abuse the traditional financial infrastructure, online and mobile payment systems, or independently distributed crypto currency systems [1].

With the daily increase of different digital frauds, especially those connected to financial fraud, there is a need for more effective tools and approaches to prevent them. Furthermore, there is also a need for fraud investigators. Recently, many traditional digital forensic investigators have had to pay more attention to financial investigations. To fill the gap and facilitate work for investigators, this paper presents STALITA, an innovative platform for the analysis of bank transactions. STALITA enables the import of large-scale heterogeneous data into a complex network represented by a Neo4j graph-based database [2]. Due to data sensitivity, it enables independent transaction inspections. Graph data and their topological properties are analysed using various graph algorithms supported by Neo4j and third-party libraries, while custom-built graph-based visual analytics allow for more in-depth investigations and the discovery of new knowledge.

Furthermore, the Law Enforcement Agency (LEA) struggles to find useful information in large amounts of heterogeneous data that can be crucial for investigations. In doing so, investigators face the following challenges:

- Most of the investigations in the LEA are related to large amounts of electronic data that must be analysed;
- Most of these data are in an unstructured or semi-structured form and various formats that require the use of special domain-specific tools for data gathering, cleaning, and parsing;
- The data can include crucial information for the investigation, but the ratio between useful information in these data and all available data is very low. Therefore, LEAs need efficient tools for knowledge discovery in data;
- Almost each investigation requires a specific domain-specific approach, including various data analysis techniques and tools.

Thus, LEAs need efficient, user-friendly analytical platforms, including basic and advanced techniques, for the discovery of useful information in a large amount of heterogeneous data related to investigations.

A new platform (STALITA) for bank transaction analysis is presented in this paper, which supports heterogeneous data sources, most of the advanced techniques for graph analysis due to the use of Neo4j technology, various types of machine learning and graph-based algorithms that can be incorporated for financial fraud investigation, and, finally, various visual analytics tools. The platform supports collaborative and uniform workflow on diverse data sources through web-based applications, user-friendly extensions (template queries) for domain-specific analyses, and a high level of customisability of visual analytics tools.

The rest of the paper is organised as follows: The related work is explained in the next section. Section 3 provides the methodology used in the proposed paper. The results achieved with the proposed platform are presented in Section 4. Section 5 concludes the paper.

2. Related Work

The aim of digital forensic analysis is to determine answers to investigation and inquiry questions: who, how, what, why, when, and where [3]. To achieve this, digital forensics is the practice of collecting, analysing, and reporting on digital data in a way that is legally admissible. Therefore, several process models were proposed for forensic investigation [4]. Academics in the forensic field held consortiums and defined a general standard digital investigation process model. This model is composed of six stages: planning, incident response, collecting data, data analysis, presentation of findings, and instance closure [4]. The National Institute of Standards and Technology described the original forensic process model. This model includes four phases: collection, examination, analysis, and reporting [5]. All those steps are possible to achieve in a reasonable time if the investigated data are in an accepted quantity and quality. Due to the growing amount of data that needs to be investigated, the authors of [6] presented a realistic implementation of the reduction of collection and processing times, as well as reducing the time needed to undertake analysis and providing investigators with evidence or actionable intelligence in a timely manner [6].

The investigation of fraud in the financial domain is restricted to those who have access to relevant data. Customer financial records are protected by law and internal policies; therefore, they are not available for most of the researchers in the area of fraud detection [7]. Due to these restrictions, cyber forensic tools are more common in other areas, such as computer forensics, network forensics, mobile forensics, and database forensics [8]. However, they still do not support processing data from multimodal data sources in a uniform way. In computer forensics, EnCase is a commercial platform for deep analysis for recovering deleted files, sorting, and reviewing files, file signature analysis, internet history review, hash value and registry analysis, timeline, and gallery review [9]. Network traffic analysis, multithreading, modularity of the input and output interfaces, port-independent protocol identification, and large-scale PCAP data analysis are the main features of Xplico. XRY is a mobile forensics tool to investigate smart phones, mobile phones, tablets, and GPS navigation systems. It is used in the areas of intelligence operations, criminal investigations,

law enforcement, and military agencies [10]. The SQLite forensics browser is a database forensics tool for creating, managing, and analysing evidence during the creation of the case. The tool has the capabilities of creating databases, scanning, and previewing database files, database indexing, adding custodian entries, searching, and exporting file formats [11]. More advanced techniques need to be used due to the restrictions mentioned in the financial domain and the inability to support heterogeneous data.

Anomaly detection in the financial domain has been studied extensively for this purpose, and most of the studies rely on statistical, artificial, and machine learning techniques [12]. To overcome the challenges associated with supervised techniques, semi-supervised and unsupervised algorithms are becoming more common [12]. Frequently used models for financial fraud are decision trees, support vector machines, logistic regressions, k-means, and k-nearest neighbour clustering [13–15]. Nowadays, deep learning detection techniques have emerged and gained importance in the last few years, demonstrating significantly better performance than other techniques in addressing financial fraud problems [12]. These include neural network architectures of various types, such as convolutional neural networks, long short-term memory networks, autoencoders, and generative adversarial networks [16]. Pourhabibi et al. presented a literature review of different graph-based anomaly detection techniques that have been studied in the published literature in the context of financial fraud [17]. The authors of [17] divided graph-based approaches into five different groups, such as community-based, probabilistic-based, structural-based, compression-based, and decomposition-based [17]. Jeong et al. [18] provided an overview of the research on data mining-based fraud detection. They showed that data mining techniques are providing great aid in financial accounting fraud detection and are applied most extensively to provide primary solutions to the problems. The authors have also classified research under a few criteria, such as data set, data mining algorithm, and viewpoint of research. The authors of [19] presented a theoretical framework to predict when and how investigators might use data visualisation techniques to detect fraudulent transactions [19]. The author Astrakhantseva in [20] proposed a supplement for classical financial analysis with a special section that analyses transactions for compliance with market conditions, identifies schematic and fictitious transactions, and determines the degree of their impact on the occurrence of property insufficiency and signs of bankruptcy [20].

Comparison of State-of-the-Art Investigation Tools

The advantages and disadvantages of state-of-the-art investigation tools and platforms for the detection and analysis of criminal acts are presented in Table 1.

Table 1. Comparison of state-of-the-art investigation tools.

Tool	Advantages	Disadvantages
i2 Analyst notebook [21]	<ul style="list-style-type: none"> • Simple data importation; • Visual graph search; • Easy graph editing. 	<ul style="list-style-type: none"> • Desktop solution with no possibilities for teamwork; • Does not support query languages (e.g., SQL, Cypher); • Lack of configurability of graph visualisation; • Limited support for various data formats; • No support for AI-based queries supporting AI algorithms; • No big data analysis.
Sentinel [22]	<ul style="list-style-type: none"> • User friendly data importation; • Supports visual graph search; • Simple graph editing, such as changing the icon and adding new nodes. 	<ul style="list-style-type: none"> • Desktop application, does not support group work, central management of user rights, or central logging of user activities; • Does not include query language; • Lack of visualisation configurability; • No support for AI-based queries supporting AI algorithms.

Table 1. Cont.

Tool	Advantages	Disadvantages
Pajek [23]	<ul style="list-style-type: none"> • Large number of advanced queries for network analysis. 	<ul style="list-style-type: none"> • Demanding for use; • Desktop solution, lack of security protocols, does not allow group work, central logging for user activities; • Lack of tool configurability.
PowerBI [24]	<ul style="list-style-type: none"> • Very intuitive and user-friendly; • Allows for the import of new visualisations. 	<ul style="list-style-type: none"> • Does not support unstructured data analysis; • Limited queries; • Lack of configurability; • Security; • Does not allow effective network analysis; • Does not support AI.
QlikView [25]	<ul style="list-style-type: none"> • CSV, Excel, and SQL import; • Setting and using filters; • Pivot tables and visualisations, • Automatic linking of tables based on the column name. 	<ul style="list-style-type: none"> • Does not support the import of unstructured data; • Does not support JSON or XML; • Does not support AI.
Geotime [26]	<ul style="list-style-type: none"> • Transparent interface for spatiotemporal data; • Data filtering by time and/or space; • Built-in functionalities over spatiotemporal data. 	<ul style="list-style-type: none"> • Limited only to data with time and geographic components; • Does not support the import of unstructured data; • Does not support network analysis functionality; • Does not support AI; • No multiuser support.

The analysis of the tools used for the investigations shows some advantages and many more disadvantages in comparison to the presented STALITA. The main disadvantage of other tools is that they are only designed for desktop systems, making group work of multiple investigators on the same investigation case difficult. The incapability of using unstructured data and advanced AI methods for graph analyses are also disadvantages. In general, the investigation of criminal acts is based to a greater extent on the search for suspicious links and patterns than on the analysis of aggregated data. Relational data models, like SQL, are very suitable for calculating aggregate values, finding trends, and filtering tables. On the other hand, graph databases are better at analysing connections between nodes (persons, current accounts, banks, and companies) and finding complex patterns, which is what criminal investigations require. These databases are more intuitive for modelling individual cases.

3. Materials and Methods

The platform presented for bank transaction analysis is composed of three main components: data import, backend Application Programming Interface (API), and custom frontend visual analytics. In the continuation, each of these main parts will be dissected and described in detail, while Figure 1 shows the overall infrastructure of the proposed STALITA platform.

3.1. Data Importation

The data import is possible in three different ways: directly by using the Neo4j Cypher query language; via Extensible Markup Language (XML); and via the Representational State Transfer (REST) supported in Neo4J. Although the Neo4j Cypher query language allows for the import of diverse data via appropriate queries, it is the least useful because most data is provided in a standard format, and the following approaches are more efficient on larger datasets. A more frequently used approach is via a user interface, which is incorporated into the visual part of the platform using XML files. The ISO 20022 Standard and bank statement CAMT.053 format are used for the XML [27]. XML importation is much more suitable, as you can import data transactions received from the banks directly. The backend API is in charge of validating the XML files and then importing data directly into

the Neo4j database. The third option is to use the REST protocol, which is a very similar approach as with XML files. The complete datasets are sent to the backend API.

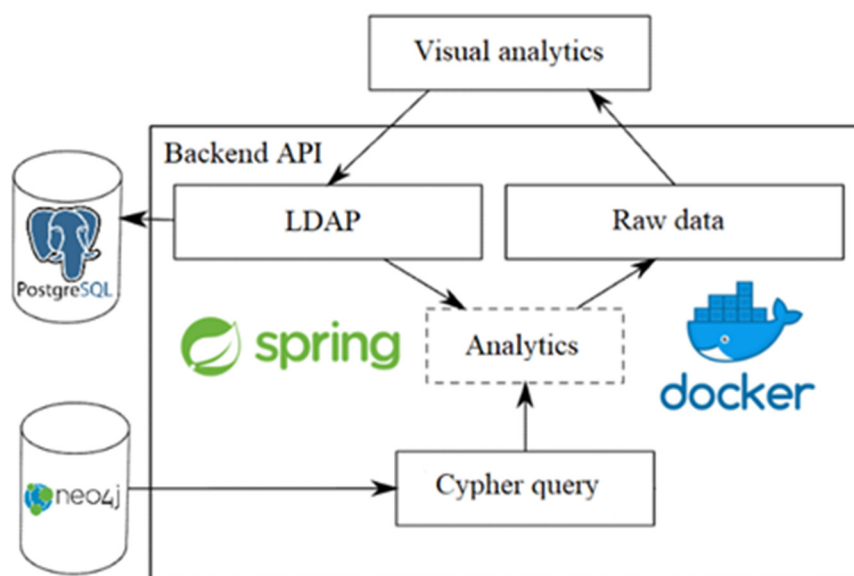


Figure 1. Infrastructure of the proposed STALITA investigation platform.

3.2. Backend API

The second main part of the proposed investigation platform is the backend API. Each of the main components, such as the Neo4j databases and the PostgreSQL database for metadata, are encapsulated in Docker containers. Docker not only enables faster deployment and testing, but also enables additional data isolation and security [28]. The latter is also significant for data from financial fraud investigations. Each investigation case is encapsulated in its own container (i.e., access is enabled only for authorised case investigators); therefore, other users have no access (e.g., unauthorised case investigators or possible intruders). Moreover, the encapsulation in containers is not the only security measure since STALITA also uses Azure Active Directory (AAD) and Lightweight Directory Access Protocol (LDAP) for user management and authentication purposes. Only the Neo4j processing part is utilised in the presented investigation platform, while the frontend visual analytics are custom made and will be described in the continuation. Neo4j's main advantage is a large set of general graph-based algorithms implemented in open-source extension libraries such as the Awesome Procedures on Cypher (APOC) and Graph Data Science (GDS) [29,30]. APOC includes over 450 standard procedures, providing functionality for utilities, conversions, graph updates, and more [29]. The metadata stored in PostgreSQL includes users and their permissions, log history, each case's metadata (e.g., case investigators, history of cypher queries, and investigator's notes), and custom entity icons are treated as the platform's metadata. The entire backend API is developed and implemented in the Java Spring framework using IntelliJ IDEA.

3.3. Frontend Application

Finally, the third main part is the web-based frontend application, which controls the backend and enables a user-friendly visual analytical task. The application runs inside a web browser, which enables high portability and execution on various platforms (from desktop environments to smart phones). The user interface is implemented in Angular, which brings a modern, portable, and responsive graphical user interface. The application supports:

- Management of users;
- Management of investigation cases;
- Data import for individual investigation cases;

- Communication between users about the specific investigation case;
- Advanced analytical tasks.

The most important part of the analytical tasks is the visualisation of graph data. The users can perform Cypher queries on specific investigation cases and visualise the results. This can be done in two ways: direct and indirect modes. In the direct mode, the user enters the Cypher query manually, which requires good knowledge of the Cypher query language. In this mode, the application increases productivity by automatically verifying the entered Cypher query and marking any potential errors. Finally, the entered query can be stored for potential reuse by other users.

For beginners, the Cypher query language is tedious to write. Therefore, the application brings a novel approach to user-friendly queries: indirect mode, which works in a way similar to user-defined extensions. This concept works as follows: An advanced user prepares a template in the Cypher query, which is an ordinary Cypher query with parametrised fields, e.g., the name of a potential suspect and date of the bank transaction. A description of each parametrised field and allowed data format are defined at the same time. Finally, the template query is saved in the backend for other users. Less advanced users can execute the template Cypher queries without any knowledge of the Cypher query language. The only task for the user is to fill in the required fields with the parameters and execute the query.

The tabular mode of visualising query results is the default, allowing for direct inspection of the Cypher query results. In the case of graph data, nodes and edges are converted into tabular form. For higher productivity, the application supports data export into the CSV format and the filtration of data.

Given that the Cypher query language works on graph-based data representation, the application supports the graphical visualisation of entities, which are represented with nodes, and connections between entities, which are represented by edges between nodes. The visualisation is performed directly inside the web browser by using 2-D hardware acceleration, which allows for interactive visualisation of large graphs with thousands of entities.

To visualise the graphs, the nodes are placed on the canvas using the force graph layout [31]. In this way, neighbouring nodes are grouped based on their connectivity. Additionally, the user can move individual nodes interactively, which moves nearby nodes proportionally to the neighbourhood. This method provides a clear visualisation of data by grouping nodes into distinct clusters. Moreover, for data with a hierarchical structure, the application can align the nodes automatically into the tree layout.

The application also supports multiple charts for data in tabular form: histograms, radial chart/flow maps, line charts, scatter plots, and pivot tables. In general, the application supports standard chart visualisation, where the user can define the columns used in the visualisation. Additionally, the application brings the following novelties to individual charts: Because a line chart is intended to be used also for temporal data, the application includes value aggregation on a daily, monthly, and hourly basis. The scatter plot is designed for discrete variables; therefore, the application supports colouring of individual entries according to the Cypher query results. In the same manner, tool-tip texts are displayed above each entry.

3.4. Definition of Graphs' Topological Properties

As was previously noted, the basic data structure of the presented platform is based on graphs. The definition of graphs, their most common topologies, and some topological properties used in graph analysis are presented in this subsection. Graphs G are defined as $G = (N, E)$, where $N = \{n_i\}$ is a node-set, and $E = \{e_{i,j}\}$ is an edge-set of G [32]. A given node, n_i , corresponds to a subject (in the context of bank transaction analysis, this could be a bank account, a person, or a company), whereas an edge, $e_{i,j} = (n_i, n_j)$, can define ownership (the person owns an account, the person owns a company), or a transaction (from one account to another). There are many different options for node and edge meanings as

we are working with unstructured data. Graph topologies are typically defined by their topological properties, and we know four major types of graphs: small-world, scale-free, random, and regular [33].

The topological properties that define graph types can also be applied to basic and assembled analyses of case investigations. Some of the properties are presented and described in the continuation:

- **Node’s degree $k[n_i]$** could, for example, be used to count a number of people’s bank accounts or transactions, and similar countings are defined as (1):

$$k[n_i] = |E_i|; \tag{1}$$

- **The average degree $k_a[n_i]$ of nodes linked to n_i** can, for example, describe the average number of transactions for the persons linked to n_i . It is defined as (2):

$$k_a[n_i] = \frac{1}{k[n_i]} \sum_{e_{ij} \in E_i} k[n_j]; \tag{2}$$

- **The number of triplets of transactions $t[n_i]$ that include node n_i** is, for example, used for the search of circles of transactions in the investigation of money laundering cases. Mathematically, it is defined as (3):

$$t[n_i] = \left| \left\{ e_{j,h} : n_j, n_h \in N_i \right\} \right|; \tag{3}$$

- **The local clustering coefficient $u[n_i]$** the proximity of a cluster of transactions, a bank account, or, eventually, people to the center. The coefficient could be written as (4):

$$u[n_i] = \frac{t[n_i]}{|E_i|(|E_i| - 1)}; \tag{4}$$

- **Local betweenness centrality $b[n_i]$** shows the size of clusters of, for example, bank accounts, persons, or transactions that a given n_i (bank account, person, or transaction) is linked with. Its mathematical definition is (5):

$$b[n_i] = \frac{1}{(|N_i| - 1)(|N_i| - 2)} \sum_{i \neq j, i \neq h, j \neq h} n_j, n_h \in N_i \frac{|\{\Pi(j, h) : n_i \in \Pi(j, h)\}|}{|\{\Pi(j, h)\}|}. \tag{5}$$

The size of nodes and edge sets in graphs also defines the time complexity for different types of graphs and different topological properties. In [28], it has been shown that the number of edges does not influence the time complexity, as they are checking all possible connections in the graph (full graph). In the case of bank transactions’ investigations, we will rarely search in full graphs, and, by considering that, we can assume that the number of nodes and edges has an influence on time complexity in our graph database structure.

4. Results

4.1. Description of the Investigation Case

This section presents an example investigation of money laundering with STALITA. The example is based on a real case involving several companies and individuals through whom money was transferred to the final recipients, who withdrew the money in cash. The starting point of the case was a notification from the Office for Money Laundering Prevention (OMLP), which informed the Law Enforcement Agency (LEA) of one suspicious transaction. A large sum of money was transferred from a certain company account to a person’s account, and the full amount was later withdrawn. The person (Person1) is the owner of the company (Company1). Based on the OMLP notification, the LEA obtained data on the turnover of the company and the person’s transaction accounts. Finally, data,

including information about senders, transactions (e.g., time, amount, and purpose), and the recipient, were inserted into STALITA.

4.2. Example Investigation Case

The purpose of the analysis was to confirm the initial findings of the OPML and determine whether this was an isolated case or not. Thus, investigators run four different STALITA tools. The first one includes a query that provides all transactions that have arrived at Person1. A cash withdrawal had subsequently been made, and it was found that the difference between the deposit and the cash withdrawal was less than or equal to 3 days. Figure 2 shows the highlighted transaction by the OMLP. The query not only confirmed the OMLP’s findings but also provided all transactions with the same pattern and all transaction accounts from which funds were transferred to Person1’s account and subsequently withdrawn in cash.

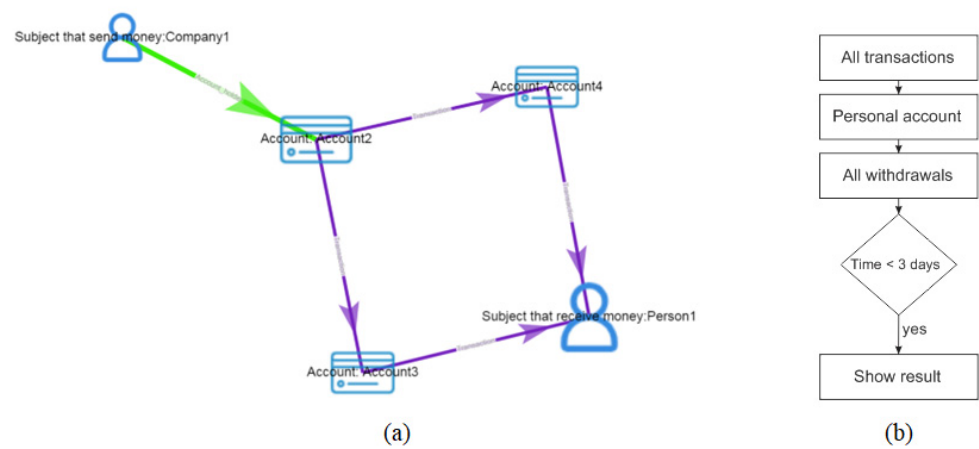


Figure 2. (a) The result of a query, where the green edge represents the initial suspicious transaction, and (b) the flowchart of the used Cypher query.

Further examination and analysis of the result of the query, which was carried out using a STALITA pivot table, revealed that more than 90% of the funds were subsequently withdrawn in cash from Person 1’s accounts. It was established that Person1 was the owner of Account3 and Account4. The next STALITA query provides all accounts from which funds have been transferred to accounts held by companies owned by Person1. The results were visualised in a STALITA graph (see Figure 3).

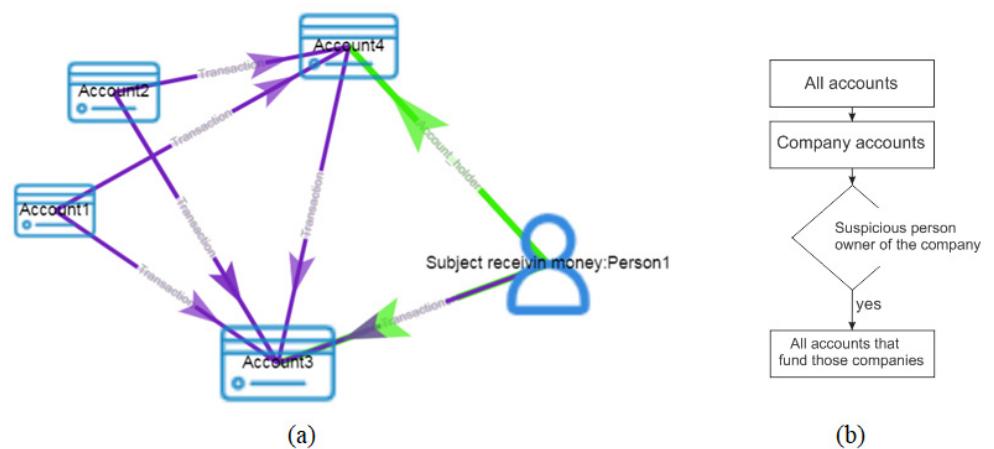


Figure 3. (a) All the accounts from which the transfers have been executed and (b) the flowchart of the Cypher query.

The next query finds all accounts that have been credited from accounts held by companies owned by Person1 (see Figure 4).

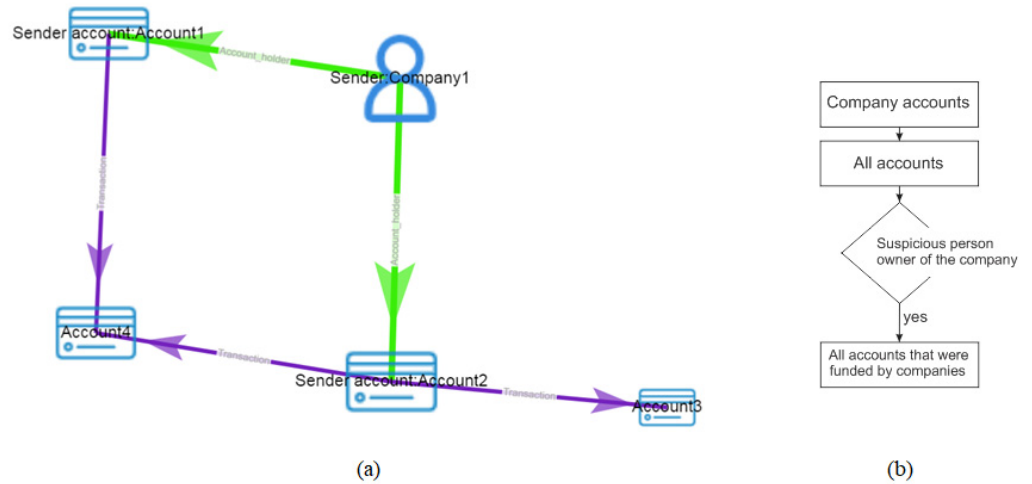


Figure 4. (a) Accounts credited from accounts held by companies' transactions and (b) the flowchart of the used Cypher query.

In the final step, new data were added to the existing data. A further purpose of the analysis was to establish the pattern of transfers between companies and individuals, thus identifying the cash flow and the real purpose of the transactions. The Cypher query searches for all transactions that start in Account2 and end in one–three steps in the account from which the cash was withdrawn, with a maximum of 5 days between the first and the last transaction, as can be seen in Figure 5.

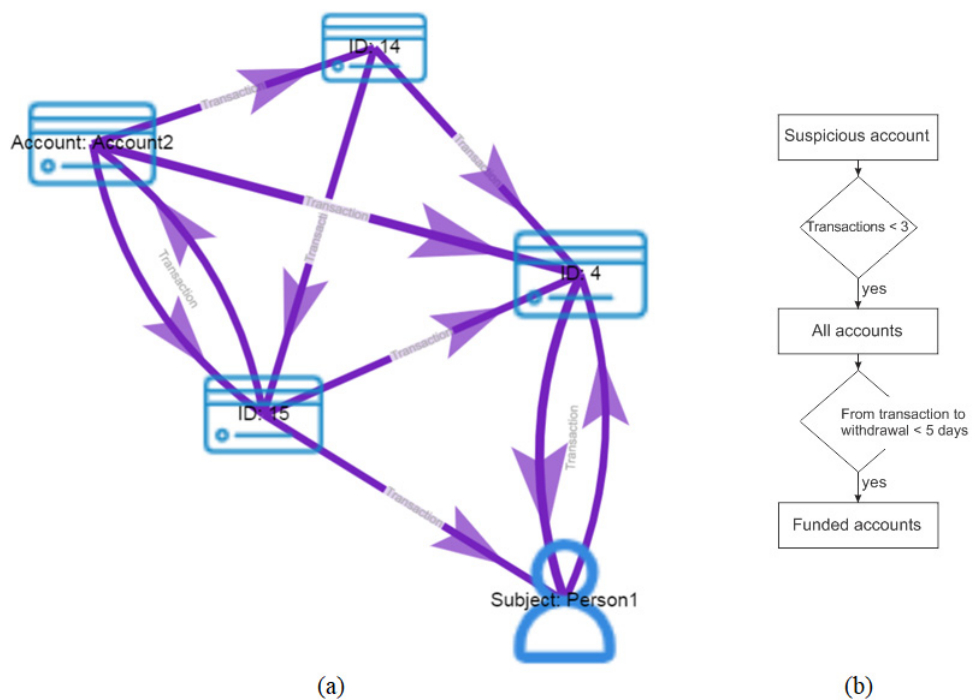


Figure 5. (a) The result of the final Cypher query, where the cycle of transactions can be seen and (b) the flowchart of the used Cypher query.

Further examination of the results in the pivot tables, time, and scatter charts shows that the highest number of cash withdrawals were made between certain days each month.

4.3. Result of the Investigation

At the end of the investigation, it was found that the companies owned by Person1 were straw companies that only served to transfer money. Individuals withdrew cash, where 10% of the amount withdrawn was retained and the rest was returned to another person (the organiser) through Person1. The organiser paid his employees the minimum wage into a transaction account and paid them bonuses, overtime, and allowances in cash to avoid having to pay taxes.

4.4. Comparison with the i2 Analyst Notebook

After the demonstration of the example investigation case in STALITA, a comparison was done with the i2 Analyst Notebook (i2 ANB) [21], which is the most used tool for similar investigations. The same data were imported into i2 ANB, and the same analyses were done. STALITA enables automated import of the CAMT XML data format, while i2 ANB does not support that functionality. Data preparation is usually the most time-consuming phase for that kind of analysis (investigations), and this is a mandatory operation for all investigations. In comparison to i2 ANB (a desktop application), STALITA is a web-based solution. In that way, STALITA enables group work, central user management, central upgrades, data insertion, and tracking of all user activities. Nowadays, these functionalities are crucial for efficient investigations of large datasets. The presented investigation platform also enables fast development of new Cypher queries that can be used as a new tool by all users. This functionality is crucial for the efficient identification of subsets of suspicious accounts and transactions in large amounts of data. On the one hand, the suspicious transactions presented in our test case can also be discovered by i2 ANB with a lot of manual work. STALITA, on the other hand, provides pre-prepared tools that advise investigators on how to analyse specific data. The chosen tool provides results in configurable tables, charts, and graphs that can be saved as templates for new visualisations. A user can configure new scatter plots (e.g., that visualise five dimensions) and save them as new recommended visualisations for certain cases/data. A similar functionality is not supported in i2 ANB. The i2 ANB is a user-friendly and intuitive tool, but it does not provide functionalities for the analysis of large datasets. Nowadays, one of the main challenges in criminal investigations is the fast analysis of large amounts of data. The first implementation of the proposed platform concept in real-life investigations shows good results that cannot be achieved by traditional tools (e.g., the i2 Analyst Notebook). The step-by-step comparison of the presented example between the proposed platform and the i2 ANB is shown in Table 2.

Table 2. Comparison of the i2 Analyst Notebook and STALITA on an example investigation case.

	i2 Analyst Notebook		STALITA	
Data importation in CAMT format	1.	Data transformation from CAMT to Excel;	1.	Using wizard in six steps to import data (select the data source and control the import).
	2.	Using wizard in 11 steps to import the data (select the source data and definition of the imported data).		
Analysis	1.	Select tab “Analyse”;	1.	Select the built-in tool (Cypher query) and enter search parameters such as the start- or end-node and keywords.
	2.	Select tool “Find path”;		
	3.	Define parameters such as entities, attributes, data ranges, and link direction.		

Table 2. *Cont.*

	i2 Analyst Notebook	STALITA
Result as pivot table	<ol style="list-style-type: none"> 1. Use the specific tool to select a graph part, copy the results to a new graph, export the data, and use it in Excel; 2. This is done in six steps: <ol style="list-style-type: none"> 2.1. Copy data to a new graph; 2.2. Open the data window to export; 2.3. Select all the data; 2.4. Copy all the data; 2.5. Paste the data into Excel; 2.6. Use the built-in pivot table. 	<ol style="list-style-type: none"> 1. Run a query using the built-in tool; 2. Select pivot table to show the results.
Time chart visualisation	<ol style="list-style-type: none"> 1. Use the specific tool to select a graph part; 2. Select “bar charts and histograms”; 3. Add the entities; 4. Select histogram aggregation type (e.g., year, month, and day). 	<ol style="list-style-type: none"> 1. Run a query using the built-in tool; 2. Select time chart visualisation; 3. Select data to show at the x- and y-axis.

5. Conclusions

Bank transactions are excellent examples of STALITA graph data analyses. Tabular data and relational databases are good for joining and aggregating but not for discovering relationships. For example, in the tables, the transactions between actors are not as visible as in the graphs. Cypher queries in STALITA are excellent tools for the discovery of suspicious patterns in the data. For example, simple queries provide answers such as: which person, in different steps, sent money to certain accounts that was withdrawn at a certain time.

An investigator can visualise results in STALITA with tables, charts, and graphs. On the one hand, a graph makes it easier to find suspicious connections. On the other hand, time charts show distributions with increasing numbers of transactions or amounts. The result can also be analysed with pivot tables and exported in tabular format. STALITA also enables the combination of bank transaction data with other data, such as phone calls. For example, investigators can supplement the database containing transactions between accounts and account holders with the telephone numbers held by these account holders and their calls to each other, allowing investigators to search for patterns and links between bank and telephone traffic.

The advantages of the platform are validated by the example investigation case presented in the results. The initial findings of the OPML have been proven by applying the tools implemented in STALITA. In four short steps of the presented investigation, the whole, usually complicated process has been completed, and the money laundering fraud has been confirmed. One of the STALITA tools presented, the Cypher query, confirmed the OPML suspicions immediately and, even more, revealed new similar accounts used for the money laundering in the same investigation case. To confirm the findings, other tools supported by the presented platform have been used, such as pivot tables, time, and scatter charts. The sums of income and the withdrawals on the suspicious accounts have been proven by applying the mentioned tools. Finally, the suspects were prosecuted criminally.

The presented investigation case is only one of many possible uses of the presented platform. The platform will be further developed in the future for specific uses with various types of data. The possible improvements to the platform are automatic data enrichment, either online or from some other sources, and support for the input of newer data formats.

Author Contributions: Conceptualisation, D.J., N.L., M.B., A.Ž., and A.P.; methodology, D.J., D.K., and Š.K.; software, D.J., D.K., M.B., and Š.K.; validation, A.Ž., A.P., and N.L.; investigation, M.B.; data curation, A.P. and A.Ž.; writing—original draft preparation, D.J., Š.K., and A.P.; writing—review and editing, N.L., B.Ž., and A.Ž.; visualisation, Š.K. and D.K.; supervision, A.P., B.Ž., and N.L.; project administration, N.L. and A.P.; funding acquisition, B.Ž., N.L., and A.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was jointly funded by the Slovenian Research Agency and the Slovenian Ministry of the Interior, grant number V2-2260. This research was also funded by the Slovenian Research Agency, grant number P2-0041.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing is not applicable due to sensitivity of the used data.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Nikkel, B. Fintech forensics: Criminal investigation and digital evidence in financial technologies. *Forensic Sci. Int. Digit. Investig.* **2020**, *33*, 200908. [CrossRef]
2. Miller, J.J. Graph database applications and concepts with Neo4j. *Proc. South. Assoc. Inf. Syst. Conf.* **2013**, *2324*, 141–147.
3. Cahyani, N.D.W.; Martini, B.; Choo, K.K.R.; Al-Azhar, M.N. Forensic data acquisition from cloud-of-things devices: Windows Smartphones as a case study. *Concurr. Comput. Pract. Exp.* **2017**, *29*, e3855. [CrossRef]
4. Oo, M.N. Forensic Investigation on Hadoop Big Data Platform. PhD Dissertation, University of Computer Studies, Yangon, Myanmar, 2019.
5. Bahuguna, A.M.H. Big Data Security—The Big Challenge. *J. Sci. Eng. Res.* **2016**, *7*, 23–32.
6. Quick, D.; Choo, K.K.R. Data reduction and data mining framework for digital forensic evidence: Storage, intelligence, review, and archive. *Trends Issues Crime Crim. Justice* **2014**, *1*, 1–11.
7. Lopez-Rojas, E.A.; Axelsson, S. A review of computer simulation for fraud detection research in financial datasets. *Future Technol. Conf.* **2016**, *1*, 932–935.
8. Vihara, F. Cyber Forensics Tools: A Review on Mechanism and Emerging Challenges. In Proceedings of the 11th IFIP International Conference on New Technologies, Mobility and Security, Paris, France, 19–21 April 2021; Volume 1, pp. 1–7.
9. Casey, E. *Handbook of Computer Crime Investigation: Forensic Tools and Technology*; Academic Press: London, UK, 2002.
10. XRY—Extract. Available online: <https://www.msab.com/products/xry/> (accessed on 9 September 2022).
11. SQLite Forensics Browser. Available online: <https://www.revove.com/database-forensics/sqlite-forensics-browser/> (accessed on 10 September 2022).
12. Hilal, W.; Gadsden, S.A.; Yawney, J. Financial Fraud: A Review of Anomaly Detection Techniques and Recent Advances. *Expert Syst. Appl.* **2022**, *193*, 116429. [CrossRef]
13. Sahin, Y.; Duman, E. Detecting credit card fraud by decision trees and support vector machines. *World Congr. Eng.* **2012**, *2188*, 442–447.
14. Jain, Y.; Tiwari, N.; Dubey, S.; Jain, S. A comparative analysis of various credit card fraud detection techniques. *Int. J. Recent Technol. Eng.* **2019**, *7*, 402–407.
15. Shpyrko, V.; Koval, B. Fraud detection models and payment transactions analysis using machine learning. *SHS Web Conf.* **2019**, *65*, 02002. [CrossRef]
16. Fiore, U.; De Santis, A.; Perla, F.; Zanetti, P.; Palmieri, F. Using generative adversarial networks for improving classification effectiveness in credit card fraud detection. *Inf. Sci.* **2019**, *479*, 448–455. [CrossRef]
17. Pourhabibi, T.; Ong, K.L.; Kam, B.H.; Boo, Y.L. Fraud detection: A systematic literature review of graph-based anomaly detection approaches. *Decis. Support Syst.* **2020**, *133*, 11303. [CrossRef]
18. Jeong, S.H.; Kim, H.; Shin, Y.; Lee, T.; Kim, H.K. A survey of fraud detection research based on transaction analysis and data mining technique. *J. Korea Inst. Inf. Secur. Cryptol.* **2015**, *25*, 1525–1540.
19. Dilla, W.N.; Raschke, R.L. Data visualization for fraud detection: Practice implications and a call for future research. *Int. J. Account. Inf. Syst.* **2015**, *16*, 1–22. [CrossRef]
20. Astrakhantseva, I.; Astrakhantsev, R. Fraud transactions revealing as phase of financial analysis in Forensic Economic Examination. *SHS Web Conf.* **2021**, *94*, 03011. [CrossRef]
21. i2 Analyst Notebook, Data-Centric Visual Analysis Tool. Available online: <https://docs.i2group.com/anb/9.4.0/> (accessed on 14 November 2022).
22. Sentinel, Sentinel Visualizer. Available online: <http://www.fmsinc.com/linkanalysis/> (accessed on 14 November 2022).
23. Pajek, Analysis and Visualization of Very Large Networks. Available online: <http://mrvar.fdv.uni-lj.si/pajek/> (accessed on 14 November 2022).
24. PowerBI, Unified, Scalable Platform for Self-Service and Enterprise Business Intelligence. Available online: <https://powerbi.microsoft.com/en-us/> (accessed on 14 November 2022).
25. QlikView, Powerful Interactive Analytics and Dashboards. Available online: <https://www.qlik.com/us/products/qlikview> (accessed on 14 November 2022).
26. Geotime, Geospatial Analysis Software. Available online: <https://www.geotime.com/> (accessed on 14 November 2022).

27. SEPA for Corporates, A Practical Guide to the Bank Statement CAMT.053 Format. Available online: <https://www.sepaforcorporates.com/swift-for-corporates/a-practical-guide-to-the-bank-statement-camt-053-format/> (accessed on 10 September 2022).
28. Docker, Docker Platform. Available online: <https://www.docker.com/> (accessed on 11 September 2022).
29. APOC, Awesome Procedures on Cypher. Available online: <https://neo4j.com/developer/neo4j-apoc/> (accessed on 11 September 2022).
30. GDC, Graph Data Science. Available online: <https://neo4j.com/product/graph-data-science/> (accessed on 12 September 2022).
31. Spring Embedders and Force-Directed Graph Drawing Algorithms. Available online: <https://arxiv.org/abs/1201.3011> (accessed on 14 September 2022).
32. Mongus, D.; Vilhar, U.; Skudnik, M.; Žalik, B.; Jesenko, D. Predictive analytics of tree growth based on complex networks of tree competition. *For. Ecol. Manag.* **2018**, *425*, 164–176. [CrossRef]
33. Jesenko, D.; Mernik, M.; Žalik, B.; Mongus, D. Two-level evolutionary algorithm for discovering relations between nodes' features in a complex network. *Appl. Soft Comput.* **2017**, *56*, 82–93. [CrossRef]