HOSTED BY
ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/jides

Rogue behavior detection in NoSQL graph databases☆

*Arnaud Castelltort**, *Anne Laurent*

LIRMM - University of Montpellier - CNRS UMR 5506, Montpellier, France

HIGHLIGHTS

- We aim at detecting rogue behaviors.
- We claim that fraud data are essentially based on graphs.
- We consider NoSQL graph databases.
- We propose fuzzy historical pattern matching.
- We provide a system that allows the user to define his own terms definitions.

ARTICLE INFO

Article history:
Received 1 March 2016
Received in revised form
27 October 2016
Accepted 31 October 2016
Published online 17 November 2016

Keywords:
Rogue behavior
Fraud rings
NoSQL graph databases
Fuzzy DSL
Approximate cypher queries

ABSTRACT

Rogue behaviors refer to behavioral anomalies that can occur in human activities and that can thus be retrieved from human generated data. In this paper, we aim at showing that NoSQL graph databases are a useful tool for this purpose. Indeed these database engines exploit property graphs that can easily represent human and object interactions whatever the volume and complexity of the data. These interactions lead to fraud rings in the graphs in the form of sophisticated chains of indirect links between fraudsters representing successive transactions (money, communications, etc.) from which rogue behaviours are detected. Our work is based on two extensions of such NoSQL graph databases. The first extension allows the handling of time-variant data while the second one is devoted to the management of imprecise queries with a DSL (to define flexible operators and operations with Scala) and the Cypherf declarative flexible query language over NoSQL graph databases. These extensions allow to better address and describe sophisticated frauds. Feasibility have been studied to assess our proposition.

Peer review under responsibility of Qassim University.

☆ Fully documented templates are available in the elsarticle package on CTAN.

* *Corresponding author.*
E-mail address: arnaud.castelltort@lirmm.fr (A. Castelltort).

http://dx.doi.org/10.1016/j.jides.2016.10.004

1. Introduction

Rogue behaviors are known to lead to important economic and political concerns. Frauds in banks and insurance companies represent billions of dollars lost every year. For instance, more than £52 billion was lost in the UK in 2013 [1]. Fraud can be detected by considering abnormal patterns in the interactions. However, these anomalies are hidden and often difficult to retrieve because of their complexity. Fraud can be committed by one or more persons. It can impact on individuals or organizations (e.g., banks).

As rogue behaviors are characterized by the interactions, graphs can thus help for retrieving frauds. Graphs are indeed recognized to play an important role within the pattern recognition field [2], thus being a key technology for retrieving relevant information. Graphs efficiently represent the relationships between objects, should they refer to persons, organizations, or scientific data (e.g., chemistry). Techniques and algorithms can be distinguished in considering the fact that they are meant to mine relevant patterns or to retrieve them.

Detection is achieved through the modelization of fraud rings which are hidden within the graph of interactions. A fraud ring is a set of connections between actors. It can be found in many fraud frameworks [1].

Although graphs have been studied since the very beginning of computer science in the so-called graph theory field [3], their integration within database management systems is more recent. Some of the first systems have been proposed with the emergence of ontologies and RDF triplets queried through SPARQL [4]. More recently, NoSQL databases have proposed efficient engines devoted to graph databases: GraphDB, Neo4J, etc. [5] compares some of these engines and points the advantages of the Neo4J system, which is the one we consider.

In this paper, we propose a framework for defining fuzzy temporal pattern matching from NoSQL graph databases. For this purpose, we first recall the basic concepts of NoSQL graph databases, temporal queries and graph pattern matching in Section 2. We then detail the problem we address in Section 3, before presenting a first attempt for addressing the problem using the NoSQL Neo4j graph database in Section 4. The proposition has been implemented. The main contribution of this paper is presented in Section 5. This contribution is mainly based on the use of generalized fuzzy queries. These queries can be user-defined and rely on a Domain Specific Language (DSL) and on an extension of the declarative query language to better address and describe sophisticated frauds. Section 6 reviews the main contributions from the literature related to rogue detection. Section 7 sums up this paper and presents the future work we would like to address.

2. Preliminary statements

2.1. Graphs

Graphs have been studied for a long time by mathematicians and computer scientists. A graph can be directed or not. It is defined as follows.

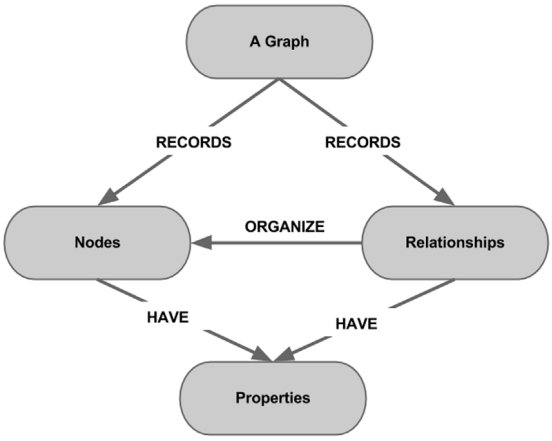

Fig. 1 – Labeled graph.

Definition 1 (*Graph*). A graph G is given by a pair (V, E) where V stands for a set of vertices and E stands for a set of edges with $E \subseteq (V \times V)$.

Definition 2 (*Directed Graph*). A directed graph G is given by a pair (V, E) where V stands for a set of vertices and E stands for a set of edges with $E \subseteq \{V \times V\}$. That is E is a subset of all ordered permutations of V element pairs.

When used in real world applications, graphs need to be provided with the capacity to label nodes and relations, thus leading to the so-called labeled graphs, or property graphs as shown in Fig. 1 and defined below:

Definition 3 (*Labeled Oriented Graph*). A labeled oriented graph G, also known as oriented property graph, is given by a quadruplet (V, E, α, β) where V stands for a set of vertices and E stands for a set of edges with $E \subseteq \{V \times V\}$, α stands for the set of attributes defined over the nodes, and β the set of attributes defined over the relations.

Given such graphs, it is possible to retrieve subgraphs, as described below.

2.2. Graph pattern matching and querying

Graph pattern matching is a very difficult algorithmic problem that has lead to the production of many works. We focus here on the definition and usage of pattern matching queries. Some related works are presented in Section 6.

The goal of pattern matching is to retrieve a pattern from data. In graph pattern matching, the pattern and the source data are both organized as graphs, as illustrated in Fig. 2.

More formally, graph pattern matching amounts to retrieve all occurrences of a graph pattern Q from a source graph G. The problem of deciding whether a subgraph is included within another one is known as subgraph isomorphism, which is known to be NP-complete.

Definition 4 (*Subgraph Isomorphism*). Let $Q = (V_Q, E_Q)$ and $G = (V, E)$ be graphs. A subgraph isomorphism from Q to G is a function $f : V_Q \rightarrow V$ such that if $(u, v) \in E_Q$, then $(f(u), f(v)) \in E$. f is an induced subgraph isomorphism if in addition if $(u, v) \notin E_Q$, then $(f(u), f(v)) \notin E$.