**PROJECT REPORT**

***on***

**Fraud Detection System using graphs**

***(****in* ***Neo4j)***

*(Btech AI&DS IV Semester Mini project)*

*(2020-24)*



***Submitted to:***

*Department of computer Applications*

***Submitted by****:*

*Rajarshi Bhattacharjee(15)*

*University Roll Number: 2017639*

*Under the guidance of* ***Mrs Garima Sharma***

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

**GRAPHIC ERA DEEMED TO BE UNIVERSITY, DEHRADUN**

***DECLARATION***

# I, Rajarshi Bhattacharjee student of Graphic Era Deemed To Be University, Dehradun, studying in **B-tech, Semester 4**, Department of Computer Science and Engineering, declare that the technical project work entitled “Fraud Detection System using graphs” has been carried out by me and submitted in partial fulfilment of the course requirements for the award of degree in B-tech of **Graphic Era Deemed To Be University** during the academic year **2021-22**. The matter embodied in this synopsis has not been submitted to any other university or institution for the award of any other degree or diploma.

***ACKNOWLEDGEMENT***

I would like to take this opportunity to express my gratitude to entire faculty at Department of Computer Science and Information Technology, Graphic Era Deemed To Be University, Dehradun who evaluated the project from time to time and guided me for making the project.

I am grateful to **Ms Garima Sharma**, Graphic Era Deemed To Be University, for her supervision, encouragement, inspiration, and guidance. Working under her is being an enriched experience. In all, I found congenial work environment in Graphic Era University, Dehradun and this project completion will mark a new beginning for me in the coming days.

I am highly indebted to Graphic Era University for providing me the required infrastructure and facilities to accomplish the given task.

Rajarshi Bhattacharjee

Btech AI&DS

2020-24

Graphic Era University

***FRAUD DETECTION on bank payements***

* ***Problem Statement***:

We can’t but agree that we stand at the beginning of a new digital payment era that payment fraud is the most common fraud type. As for the online business and banks payment fraud detection is now becoming one of the main issues. Fraud is a billion-dollar business, and it is increasing every year. There are millions of transactions that are taking place through banks, such transactions cannot be monitored and authenticated individually, because of this problem there are huge amount of money that is lost. In times of recession and economic crisis, payment fraud tends to increase, due to the Covid-19 pandemic, which has had a significant impact on the financial sector. No surprise that many scammers have become more active and, as a result, payment fraud detection has become one of the most significant problems.

Even if fraud seems to be scary for businesses it can be detected using intelligent systems such as rules engines or machine learning. A rules engine is a software system that executes one or more business rules in a runtime production environment. These rules are generally written by domain experts for transferring the knowledge of the problem to the rules engine and from there to production. Rules are great for detecting some type of frauds, but they can fire a lot of false positives or false negatives in some cases because they have predefined threshold values.

For these type of problems ML comes for help and reduce the risk of frauds and the risk of business to lose money. With the combination of rules and machine learning, detection of the fraud would be more precise and confident. Two important goals to achieve are: **predictive analysis** to stop a fraudulent action before it occurs and find**recurrent patterns** and understand**client behaviours** from data.

* ***How a graph database can fit into fraud detection :***

When we talk about searching recurrent patterns and behaviours we are also talking about relationships. If we consider a fraud, we are taking in account different subjects that are linked together in some way. Those relationships can give us valuable insights of our data. When we need to represent entities or subjects and their relationships the best fit is a graph.

Searching a pattern using a graph means searching a subgraph but not only, recurrent client behaviours can be viewed as subgraphs as well. Discovering “strange” behaviours or anomalous links among subjects requires us to overcome the computational complexity associated with the traversal of data relationships.

Whether we are building an automated fraud detection system that can detect and prevent fraud as it occurs or we are providing a tool to our analysts to help with manual fraud detection, real-time traversal of a complex and highly interconnected dataset is essential.

Another important aspect to take in account is that traditional fraud analytics looks for outliers but **fraudsters try to act normal to avoid detection**. What we need is to detect fraudulent links also analysing normal behaviours.

* ***Why Neo4j?***

Traditional fraud prevention measures focus on discrete data points such as specific accounts, individuals, devices, or IP addresses. However, today’s sophisticated fraudsters escape detection by forming fraud rings comprised of stolen and synthetic identities. To uncover such fraud rings, it is essential to look beyond individual data points to the connections that link them.In Neo4j, the transactions are stored as a graph where related pieces of data are connected making it easy to traverse those relationships in real time and to find the fraudulent patterns quickly.

Neo4j’s**versatile property graph model** makes it easier for organizations to evolve fraud detection data models and rules, helping security teams match the pace of ever-advancing fraudsters. Neo4j’s **native graph processing engine** supports high-performance graph queries on large datasets to enable real-time fraud detection. The built-in, high-availability features of Neo4j ensure your mission critical fraud detection applications are always available.

***Fraud Detection model:***

We detect the fraudulent transactions from the Bank history datasets. This synthetically generated dataset consists of payments from various customers made in different time periods and with different amounts. We will apply this operations on the model:

1.Exploratory Data Analysis (EDA): In this we will analyse how the transactions are represented in the table

2.Data Pre-processing: In this part we will pre-process the data and prepare for the training. **Fraud data** will be imbalanced, to balance the dataset one can perform oversample or undersample techniques. Oversampling is increasing the number of the minority class by generating instances from the minority class . Undersampling is reducing the number of instances in the majority class by selecting random points from it to where it is equal with the minority class

3.Oversampling with SMOTE: We will perform an oversampled technique called SMOTE (Synthetic Minority Over-sampling Technique). SMOTE will create new data points from minority class using the neighbour instances so generated samples are not exact copies but they are similar to instances we have. We perform SMOTE for increasing the accuracy and the overall performance of the model

4.K-Neighbours Classifier & Random Forest Classifier: We use these for evaluating the data sets as fraud and non fraud. They are necessary for classifying, as they will decide the authenticity of the payment and thus save from fraudential activities. These two classifiers also gives there individual performance, thus giving the better working classifier in the dataset.

We use neo4j for analysing and establishing relations between the data so that we can identify the common relations that are used by frauds. This relations are necessary for classifying as fraud payment and non fraud payment

***Conclusion***

Following a trail looking for connections among entities is a very fast operation in [Neo4j](http://www.neo4j.com/). This technology allow to ask sophisticated questions about the connections in our data. Thanks to its fast performance when walking the graph, Neo4j also enables real-time detection preventing frauds before they happen.

Our model can evolve easily and become even more complex in the future integrating other internal or external sources. Regardless the complexity of our model the queries are clear, concise and fast.  Since fraud datasets have an imbalance class problem we performed an oversampling technique called SMOTE and generated new minority class examples.

Modern fraud detection tools can improve by looking beyond individual data points to connection that link them. The best solution to do this is a native graph database like Neo4j.