Analysis of Fraudulent in Graph Data base for Identification & Prevention

Deepak Singh Rawat
Research Scholar
Department of Information Technology
G.B. Pant University of Agriculture and Technology, Pantnagar, Uttarakhand, India

Rajesh Shyam Singh
Assistant Professor
Department of Information Technology
G.B. Pant University of Agriculture and Technology, Pantnagar, Uttarakhand, India

Dr. H. L. Mandoria
Professor
Department of Information Technology
G.B. Pant University of Agriculture and Technology, Pantnagar, Uttarakhand, India

Abstract

The world is changing, and this is the digital era. Almost everything around us is digitized and the flow of information is huge from a variety of sources ranging from mobile phone, smart devices, surveillance, sensors of the universe, weather forecasting sensors, medical equipment, customers transactions of the internet, user behaviours on the internet, and so on. Billions of dollars get wasted every year due to fraud. Traditional methods of fraud detection play an important role in minimizing these losses. Increasingly fraudsters have developed a variety of way to elude their detection, both by working together and by leveraging various other means of constructing fake identities. This paper proposed a new approach for fraud prevention in different sector with help of graph database by identifying of previous fraud records in graph database.

Keywords: Graph Database, Graph DB, Graph Dataset, Frauds, online Frauds

I. INTRODUCTION

Graph databases (GDB) are now a new option to Database Management Systems (DBMS). It is being used in various fields like Science, biological science, semantic web and long-range informal communication (social network). Database that hold onto connections as a central part of their information model can store, process, and inquiry associations more proficiently. A graph database stores associations and grants you to quickly cross an extensive number of associations and relations inside a little measure of time. The property graph incorporates related parts (the nodes) which can maintain the characteristics (key-value sets). A node can be named with a mark by addressing in the graph database. In a graph database, every record must be analyzed independently amid an inquiry keeping in mind the end goal to decide the structure of the information. For some reasons specialists are moving towards graph database, some of them are much speedier than previous graph databases. A graph database administration system (henceforward, a graph database) is actually online database management system along with Create, Read, Update, and Delete (CRUD) strategies which show a graph data model. Graph database tends to be commonly made for the usage alongside transactional systems. Appropriately, they are included most frequently to enhance the transactional efficiency and while designing the focus are on transactional stability and functional accessibility.

OUNCE OF PREVENTION = POUND OF CURE

A. Graph Database

In processing, a graph database is a database that utilizations graph structures for semantic inquiries with nodes, edges and properties to speak to and store information. A key idea of the framework is the graph (or edge or relationship), which straightforwardly relates information things in the store. A Graph database spares data utilizing a diagram, the most basic of information systems, appropriate for traditionally showing any kind of data in a to a degree effortlessly available means.

"A Graph Database —manages a→ Graph and —also manages related→ Indexes"

B. Nodes and Relationships

The minimum troublesome possible graph is alone Node, a record which incorporates imply to as qualities said to even as attribute. A Node starts with a single Property and creates to two or three million, that can get to some degree ungraceful. Consequent, it confronts perfect to expand the data into various nodes, dealt with express Relationships. *(fig1.1)*

C. Relations classify the Graph

Associations orchestrate Nodes inside total systems, empowering a Graph with a specific end goal to seem like a record, a Tree, a diagram, or a compound association – all of which is generally mixed inside yet more convoluted, and high between associated structures.

D. Traversal in a Graph

The Traversal is in reality exactly how we question a Graph, driving from beginning up Nodes towards related Nodes comparing with a calculation, finding reactions to inquiries like "what sound will my mates like in which I don't at present acquire," or "if this specific power supply tumbles off, precisely what web arrangements have a tendency to be affected?". *(fig.1.2)*

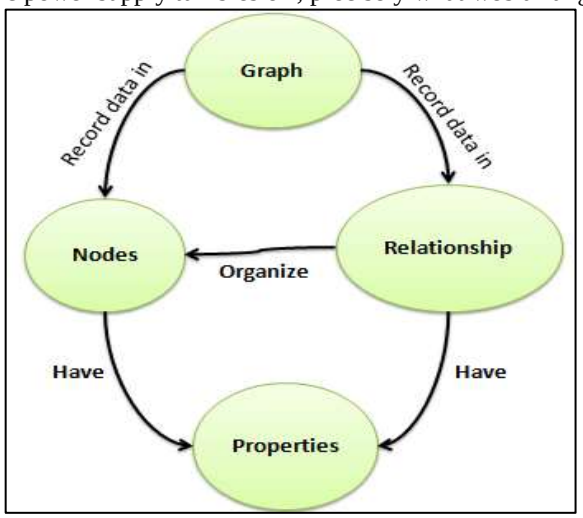

Fig. 1.1: Relations organize the Graph

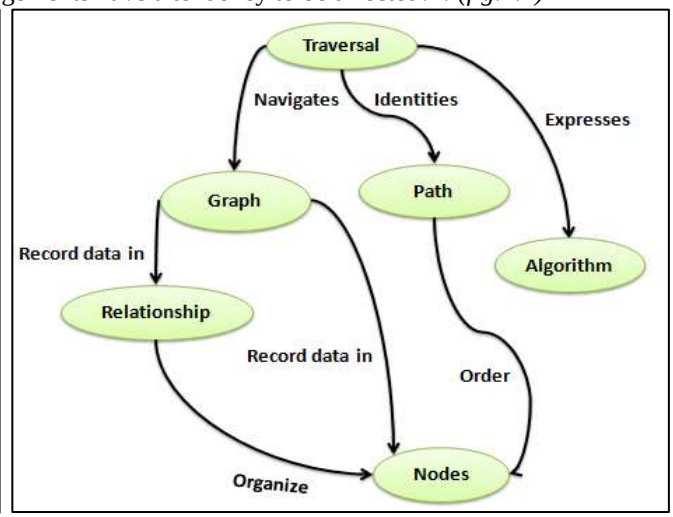

Fig. 1.2: Display Query a Graph with Traversal

E. Neo4j

Neo4j is a business reinforced open-source graph database. It was really created and furthermore created starting from the earliest stage to be dependably a tried and true database, improved for the graph systems on the other hand of tables. Performing with Neo4j, your product turns into all the expressiveness associated with a chart, nearby the majority of the steadiness you foresee out of a database.

F. Graph Compute Engines

A graph calculates system is actually a system that makes it possible for graph computational algorithms to generally be operated in opposition to great datasets. Graph compute engines tend to be developed to-do things such as recognize groups within your information, or reply queries such as, "exactly how many interactions, an average of, really does everybody inside a social network have actually?"

graph compute engines tend to be usually enhanced concerning checking and handling big quantities of data in order, plus in that respect they tend to be comparable to different order evaluation systems, such as data mining and OLAP, which is acquainted around the relational community. Although a few graph compute engines consist of a graph storage space layer, other individuals (and perhaps most) worries by themselves solely using handling information that is certainly provided inside including an exterior , as well as coming back the results. The structure consists of a method of document database with OLTP attributes (such as MySQL, Oracle, or Neo4j), which assists, needs, and acts to inquiries that you got from the program (and eventually the users) at runtime. *(fig.1.10)*

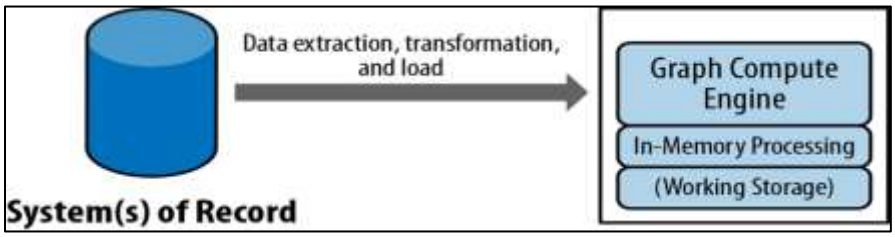

Fig. 1.3: Graph computes engine deployment

G. gSpan Algorithm

Graph-Based Substructure Pattern Mining that introduced gSpan algorithm which usually finds out regular substructures without having candidate production. gSpan develops a new lexicographic arrangement among the graphs ,and routes every graph to a exclusive minimum DFS code as the canonical label. Dependent upon this lexicographic order, gSpan explores the depth-first

search approach to exploit regular connected subgraphs effectively. So, gSpan outperforms FSG by the order of degree as well as is suitable to exploit huge regular subgraphs in a larger graph arranged with lower minimal helps.

1) *GraphSetProjection(D,S).*

```
1)  arrange the labels in D by their regularity;
2)  eliminate occasional vertices and edges;
3)  relabel the leftover vertices and edges;
4)  S1← all regular 1-edge graphs in D  ;
5)  sort S1 in DFS lexicographic order;
6)  S ←S1
7)  for every edge e € S1 do
8)  initialize s alongside e, set S. D  by graph which includes e
9)  SubgraphMining( D,S,s);
10) .D←D-e
11) if │D│< min Sup
12) break;
```

2) *Subprocedure 1 SubgraphMining(D,S,s)*

```
1)  if s ≠ min(S)
2)  S←S U
3)  specify s in every graph in D and count its children;
4)  for each c, c is s' child do
5)  if support (C) > min Sup
6)  s←c
7)  SubgraphMining(D,S,s_);
```

H. Graph Optimization Process

There are a few procedures to accomplish the enhancement of regular subgraphs in graph mining. PSO optimization based methodology is utilized to accomplish the desired results. In this thesis we exhibit a correlation between the outcomes accomplished as far as subgraphs. The correlation is between the quantity of subgraphs recognized when a looking strategy is connected on the graph database and when the PSO optimization based methodology is connected to the graph database. The pattern distinguished and the distinction regarding the number of subgraphs is of value. Particle Swarm optimization (PSO) exploits a comparative system for taking care of optimization issues. The algorithm of PSO seeks from the behavior of animals societies that don't have any chief in their group, like as bird flock and fish community.(PSO) takes motivation from the behavior of some animal societies. There are a few systems to accomplish the advancement of continuous subgraphs in graph mining. Particle Swarm Optimization based methodology is utilized to accomplish the desired results. The comparison is anywhere between the quantities of subgraphs recognized whenever a searching strategy is practiced upon the graph database as well as whenever the Particle Swarm Optimization dependent strategy is utilized towards the graph database. This particular enhancement is of perfectly Relevance to the program. Particle swarm Optimization algorithm (PSO) is basically a method formulated on agents who imitate the all-natural actions of ants, and this includes systems of collaboration and adjustment.

Each particle tries to modify its current position and velocity according to the distance between its current position and pbest, and the distance between its current position and gbest.

I. Update particles' velocities:

Move particles to their new positions:

$$\text{Current Position}[n+1] = \text{Current Position}[n] + v[n+1]$$

v_{n+1}: Velocity of particle at n+1 th iteration
V_n : Velocity of particle at nth iteration
c_1 : acceleration factor related to gbest
c_2 : acceleration factor related to lbest
rand1( ): random number between 0 and 1
rand2( ): random number between 0 and 1
gbest: gbest position of swarm
pbest: pbest position of particle
Current position[*n+1*]: position of particle at *n+1th* iteration
Current position[*n*]: position of particle at *nth* iteration *v* [*n+1*]: particle velocity at *n+1th* iteration

1) *Algorithm:*

```
For each particle
{
Initialize particle
}
```

Do until maximum iterations or minimum error criteria
{
For each particle
{
Calculate Data fitness value
If the fitness value is better than pBest
{
Set pBest = current fitness value
}
If pBest is better than gBest
{
Set gBest = pBest
}
}
For each particle
{
Calculate particle Velocity
Use gBest and Velocity to update particle Data
}

The gBest value only changes when any particle's pBest value comes closer to the target than gBest. Through each iteration of the algorithm, gBest gradually moves closer and closer to the target until one of the particles reaches the target.

II. RESULTS

Results concluded with proposed approach implementation with graph dataset. We optimize dataset with all results which are identified from the previous analysis with the help of particle swarm optimization technique. Present results of implementation without optimization and with optimization.

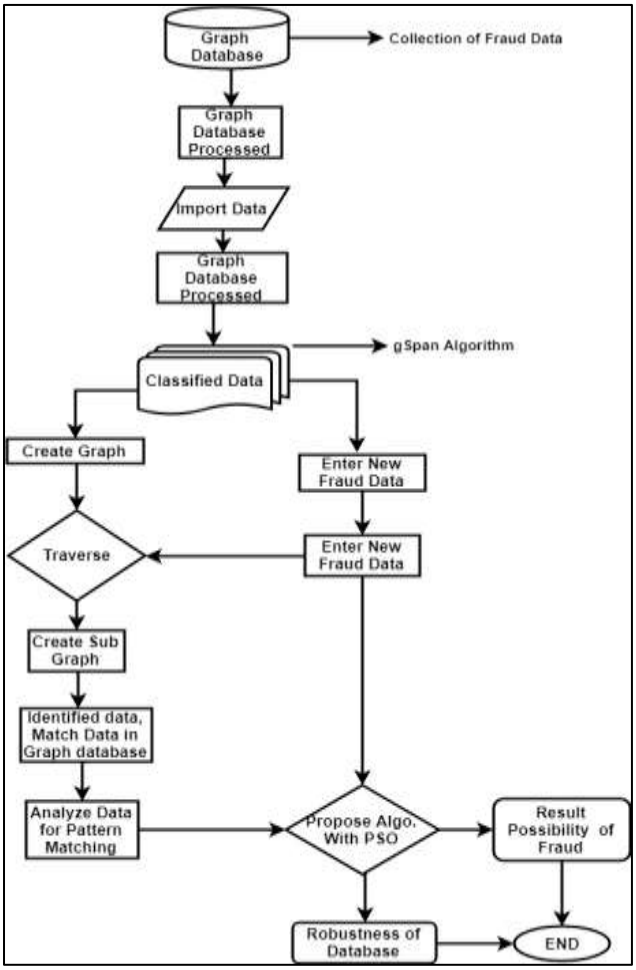

Fig. 1.4: Flow diagram of working process

A. Base analysis results

Results concluded with base approach implementation through graph dataset. We define the dataset with every one of results which are identified and displayed with graphical representation of previous results.

All these results utilize level and mode of operation with case two. Here we essentially see the variation after apply PSO approach with optimized result.

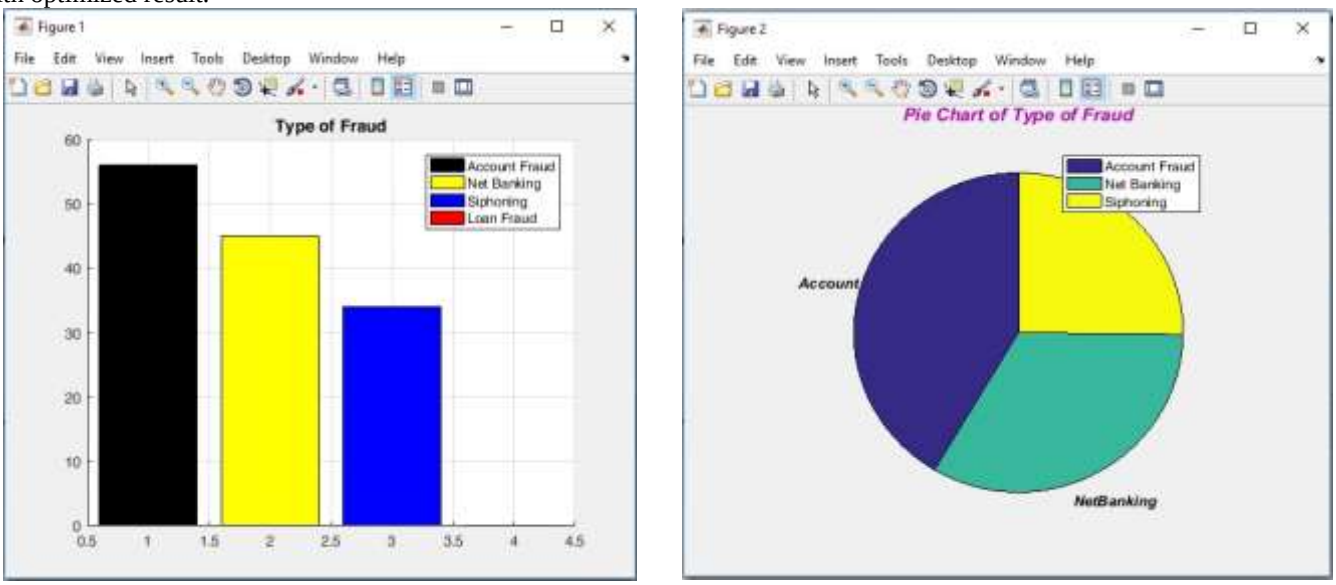

Fig. 1.5: (a), (b) Bar graph and pie chart of type of fraud detected

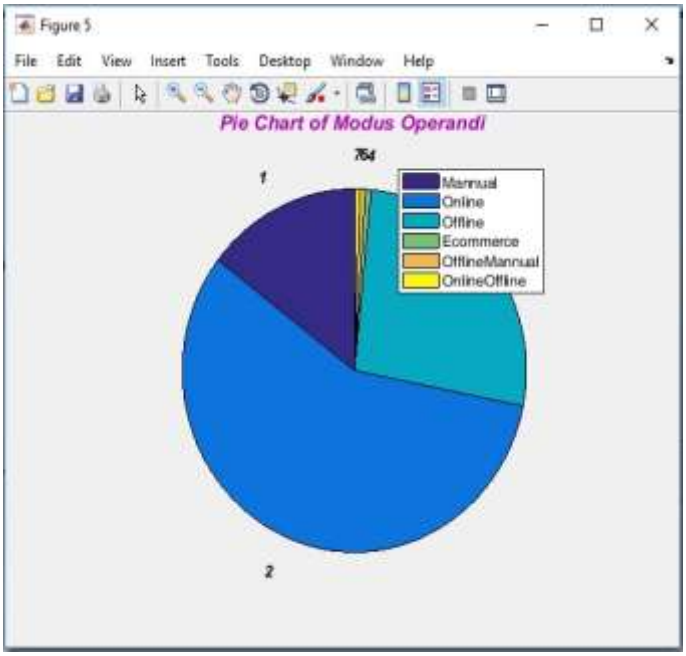

Fig. 1.6: (c) Pie chart of fraud detected with mode of operation

B. Proposed analysis results

Results concluded with proposed approach implementation through graph dataset. We optimize dataset with all results which are identified from the previous analysis with the help of particle swarm optimization technique. Present results of implementation with optimization.

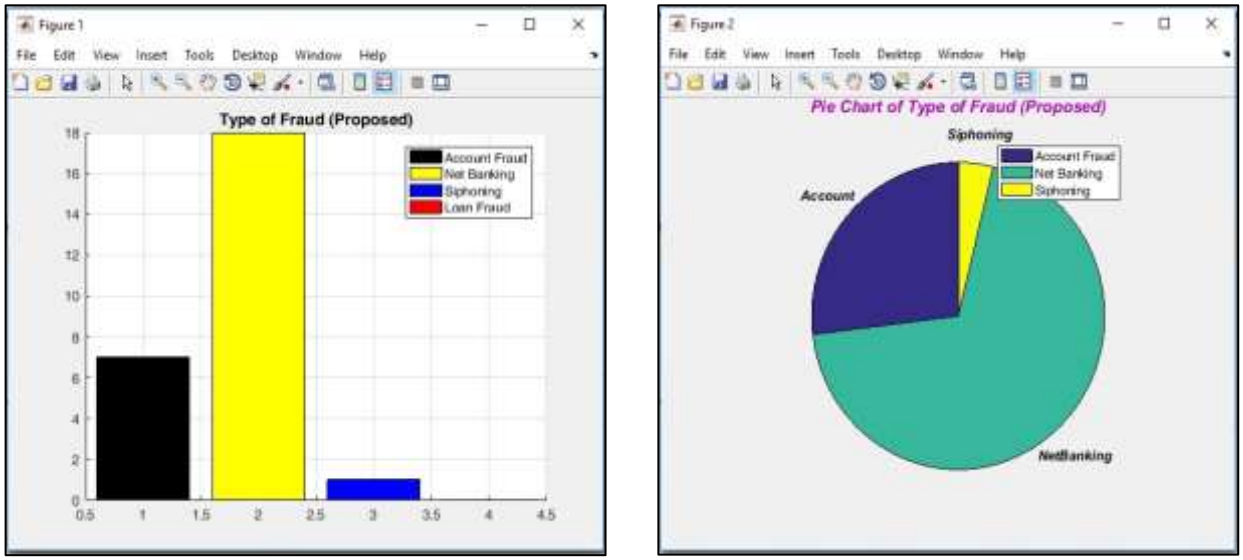

Fig. 1.7: (a), (b) Bar graph and PIE chart of type of fraud detected

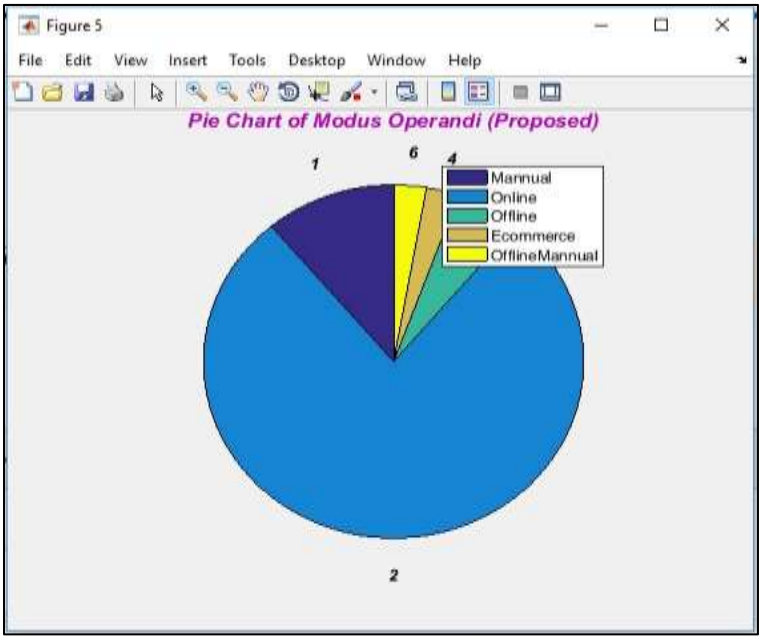

Fig. 1.8: (c) Pie chart of fraud detected with mode of operation

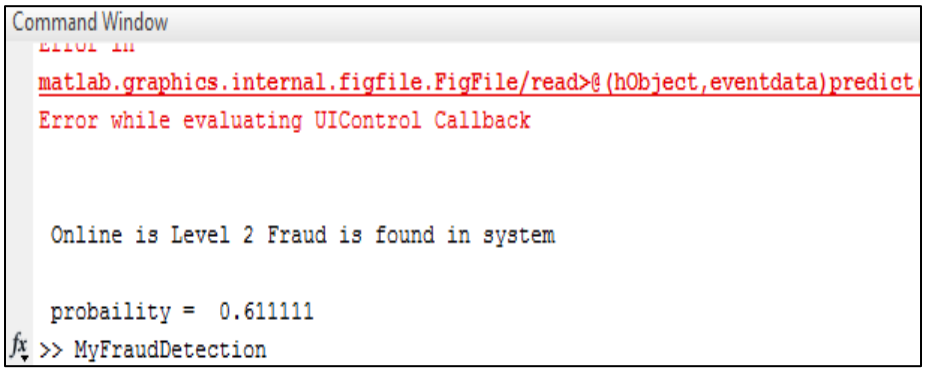

Fig. 1.9: Show Result if Fraud Detected

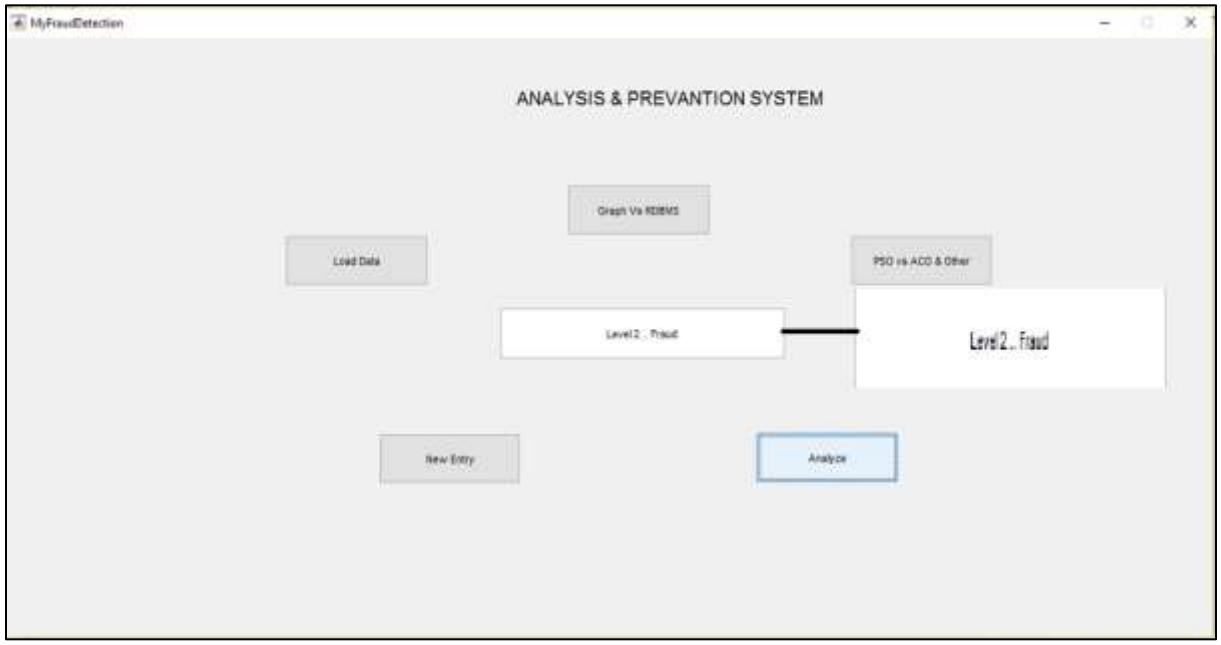

Fig. 1.10: Fraud Detected

III. CONCLUSION

We proposed an inventive calculation those arrangements utilizing the gigantic database fusing the administrations which records the properties in the diagram in a few criteria and look at the association between them in at the same time left and also right way, in this way receiving DFS system. It moreover finds the subgraph through crossing the graphical record and pulling the required example. The proposed calculation is really connected for acknowledgment of extortion and peculiarities through gathering the properties and deciding the connections which may potentially exists in the middle of the individual participating in that specific misrepresentation, modus operand which diminishes various misrepresentation which may perhaps occur in not so distant.

REFERENCES

[1] Agrawal, R. and Srikant, R., 1994 "Fast Algorithms for mining association rules. In the proc. Of the 20th Int. conf. on very large databases (VLDB), 1994.
[2] Angles, R. and Gutierrez, C., 2008. "Survey of graph database models." ACM Computing Surveys (CSUR) 40.1
[3] Angles, R., 2012. "A comparison of current graph database models." Data Engineering Workshops (ICDEW), IEEE 28th International Conference on. IEEE.
[4] Artis, M., Ayuso M. & Guillen M. 1999. "Modeling Different Types of Automobile Insurance Fraud Behaviour in the Spanish Market". Insurance Mathematics and Economics 24: 67- 81.
[5] Asai, T., Abe, K., Kawasoe, S., Sakamoto, H., Arimura, H. and Arikawa, S., 2004. "Efficient substructure discovery from large semi-structured data." IEICE TRANSACTIONS on Information and Systems, 87(12), pp.2754-2763.
[6] Barse, E., Kvarnstrom, H. & Jonsson, E., 2003. "Synthesizing Test Data for Fraud Detection Systems." Proc. of the 19th Annual Computer Security Applications Conference, 384-395.
[7] Bhardwaj, V. and Johari, R., 2015. "Big data analysis: Issues and challenges" GGSIPU, USICT, Delhi, India ISBN: 978-1-4799-7676-8
[8] Bolton, R. & Hand, D., 2001. "Unsupervised Profiling Methods for Fraud Detection". Credit Scoring and Credit Control VII.
[9] Bogdanov, P., 2008. "Graph searching, indexing, mining and modeling for Bioinformatics, cheminformatics and Social network." IEEE.
[10] Cortes, C. & Pregibon, D. 2001. "Signature-Based Methods for Data Streams." Data Mining and Knowledge Discovery 5: 167- 182.
[11] Cahill, M., Chen, F., Lambert, D., Pinheiro, J. & Sun, D., 2002. "Detecting Fraud in the Real World." Handbook of Massive Datasets 911-930.
[12] Chiu, C. & Tsai, C., 2004. "A Web Services-Based Collaborative Scheme for Credit Card Fraud Detection." Proc. of 2004 IEEE International Conference on e-Technology, e- Commerce and e- Service.
[13] Chen, C.C., Lee, K.W., Chang, C.C., Yang, D.N. and Chen, M.S., 2013. "Efficient Large Graph Pattern Mining for Big Data in the Cloud", Big Data, 2013 IEEE International Conference on, 6-9 Oct. 2013, INSPEC Accession Number: 13999251, IEEE.
[14] Di Fatta, G. and Berthold, M.R., 2005. "High Performance Subgraph Mining in Molecular Compounds", HPCC 2005, pp 866-877
[15] De Virgilio, R., Maccioni, A. and Torlone, R., 2013. "Converting relational to graph databases." In: SIGMOD Workshops - GRADES
[16] Duggal, P.S. and Paul, S., 2013. "Big Data Analysis : Challenges and Solutions", international Conference on Cloud, Big Data and Trust , Nov 13-15, RGPV.
[17] Gupta, R., Gupta, S. and Singhal, A., 2014. "Big data: overview." arXiv preprint arXiv:1404.4136.
[18] Huan, J., Wang, W., Bandyopadhyay, D., Snoeyink, J., Prins, J. and Tropsha, A., 2004. "A. Mining Sapitial Motifs from Protein Structure Graphs." In Proc. 8th int. conf. Research in computational Molecular Biology (RECOMB), pp. 308-315.
[19] Huan, J., Wang, W. and Prins, J., 2003. "Efficient mining of frequent Subgraph in the Presence of Isomorphism." In Proc. 2003 int. conf. Data mining (ICDM'03), pp. 549- 552.
[20] Hert, M., Reif, G. and Gall, H.C., 2011. "A comparison of rdb-to-rdf mapping languages." In First International Workshop on Graph Data Management Experiences and Systems (p. 1). ACM.
[21] Jasper, 2011. "A Survey on Graph Databases" 'Jasper Tech Blogs 2011-11-25 jasperpeilee.wordpress.com.
[22] Kokkinaki, A., 1997. "On Atypical Database Transactions: Identification of Probable Frauds using Machine Learning for User Profiling." Proc. of IEEE Knowledge and Data Engineering Exchange Workshop, 107-113.
[23] Ketkar, N.S., Holder, L.B. and Cook, D.J., 2009. "Empirical Comparison of Graph Classification Algorithms", In Computational Intelligence and Data Mining, 2009. CIDM'09. IEEE Symposium on (pp. 259-266). IEEE.

[24] Kim, J., Ong, A. & Overill, R., 2003. "Design of an Artificial Immune System as a Novel Anomaly Detector for Combating Financial Fraud in Retail Sector". Congress on Evolutionary Computation.
[25] Karunaratne, T. and Boström, H., 2007. "Using background knowledge for graph based learning: a case study in chemoinformatics." Springer, Artificial Inteligence, (6), pp. 151-153.
[26] Le, T.V., Kulikowski, C.A. and Muchnik, I.B., 2008. "Coring Method for Clustering a Graph", In proceedings of IEEE
[27] Moreau, Y., Lerouge, E., Verrelst, H., Vandewalle, J., Stormann, C. & Burge, P., 1999. "BRUTUS: A Hybrid System for Fraud Detection in Mobile Communications. Proc. of European Symposium on Artificial Neural Networks, 447-454.
[28] Motoda, H., 2006. "What Can We Do with Graph-Structured Data A Data Mining Perspective" Springer 2006, pp 1-2.