Arabian Journal for Science and Engineering
https://doi.org/10.1007/s13369-021-05682-9

RESEARCH ARTICLE-COMPUTER ENGINEERING AND COMPUTER SCIENCE

Credit Card Fraud Detection Technique by Applying Graph Database Model

Debachudamani Prusti[1] · Daisy Das[1] · Santanu Kumar Rath[1]

Received: 4 April 2020 / Accepted: 20 April 2021
© King Fahd University of Petroleum & Minerals 2021

Abstract
Digital transactions using credit cards are observed to be increasing day by day because of the convenience in operation. It is a matter of great concern for credit card users as well as financial institutions, providing credit card facilities for making the transactions free from possible frauds being carried out by fraudsters. The fraudsters apply different methodologies and alter their behaviours to undertake the fraudulent activities in both online and offline mode with some advanced techniques. Hence, developing a fraud detection system to identify the fraudulent activities is an important area of research to improve the credibility of credit card-based digital transactions. In this study, a fraud detection system has been proposed based on application of graph database model. The graph features being extracted using Neo4j tool are incorporated with several other features of transaction database. Subsequently, five supervised and two unsupervised machine learning algorithms are applied to them in order to detect fraudulent transactions explicitly. The features directly obtained from the transactional data are also tested with the classification models for detecting the fraudulent transactions. Critical assessment for performance of the machine learning algorithms has been carried out based on the features extracted from graph database and features extracted directly from the transaction database.

Keywords Credit card fraud detection · Graph feature · Graph database model · Neo4j tool

1 Introduction

Credit card is a financial tool that enables its holders to make transactions on credit from associated financial institutions. But there are occurrences of a good number of fraudulent activities while making both online and offline transactions using credit cards. Thus, fraudulent activities invite loss to the users as well as to the service providers. The financial institutions in the long run lose goodwill in the society. Hence, a good number of researchers in the domain of economics, finance and computer science show interest to detect the fraudulent transactions in credit card in order to save the users and industry as well. The researchers find many challenges in credit card fraud detection [1–3], which may be identified as follows:

- In order to analyse the financial transaction data of various fraudulent activities associated with a financial institution, the data are mostly of proprietary ones, so the researchers find it difficult to access.
- As compared to authentic transactions, the number of fraudulent transactions is very less, which makes the detection of fraud difficult and imprecise.
- There is no standard evaluation metric to compare the impact of fraud detection system.
- It is observed that the occurrences of fraudulent activities vary with their complexity. So, the reason of each occurrence becomes a point of concern for researchers.

The fraudsters follow different types of unusual methods to make fraudulent activities in the credit card transactions with creative ideas [4]. They are as mentioned below:

✉ Debachudamani Prusti
517cs1018@nitrkl.ac.in

Daisy Das
715cs1056@nitrkl.ac.in

Santanu Kumar Rath
skrath@nitrkl.ac.in

[1] Department of Computer Science and Engineering, National Institute of Technology, Rourkela 769008, India

Published online: 04 May 2021

1.1 Traditional Card-Related Fraud

In this type of fraud, the important information of credit card is being compromised. It includes stealing of the credit card number, name and security pin of a person and using it for fraudulent transactions [4]. The fraudster impersonates as the original user of the card to do the transaction. In some cases, the card is hijacked by fraudsters before reaching the customer's destination. This type of fraudulent transactions can be detected conveniently without any difficulties.

1.2 Merchant-Related Fraud

This type of fraud is initiated either by the owner of merchant establishments or by their untrusted employees [4]. Merchant collusion occurs when the owner or employee tries to collect credit card information of their customers and cunningly pass them to fraudsters. Another type is triangulation, where fraudsters create phishing websites and display items at heavy discount to attract the customers. When the customer buys those items with their valid credit card details, the fraudsters intercept the card credentials to commit credit card fraud.

1.3 Internet-Related Fraud

This is a very simple and easy way for fraudsters to commit fraud online and yet not getting detected [4]. Various methods being adopted are site cloning, false merchant sites, credit card generators, etc. It uses the Luhn algorithm to generate valid credit card combinations [5].

1.4 Other Varieties of Fraudulent Activities

It includes the use of stolen or lost cards, account takeover, cardholder-not-present (CNP), skimming, phishing, etc., where the fraudster applies different fraudulent techniques [4]. Skimming is a process by which the actual data on the card's magnetic strip is copied to another with an electronic magnetic card reader. Phishing is used to steal the personal information to carry out credit card fraudulent transactions. The characteristics of an improved fraud detection method are mentioned below [6]:

- Detection of fraudulent activities in a better and accurate manner.
- Detection of fraudulent activities at earliest possible time.
- Avoiding confusion to trace a fraudulent activity (Sometimes a genuine transaction is wrongly identified as a fraudulent transaction and it should be avoided).

Research Motivation

Kavitha et al. have presented the application of graphs in credit card fraud detection and other financial as well as insurance-related fraud detections [7]. Sadowski et al. have mentioned, how to discover fraud connections by using graph databases. [8]. Eifrem et al. have also discussed on, whether the usage of graph databases is a foolproof methodology to detect financial fraud [9].

Thus, it is observed that the use of graph database may turn out to be a helpful technique in the detection of financial fraud. Further, improvement of the performance can be gained by applying machine learning classifiers [10], when some graph features are incorporated into the current dataset as additional features are necessary to know if graph databases are to be used for future work, or not.

Objective of the Study

The performance metrics of the machine learning classification algorithms are evaluated initially without incorporation of any graph features. The objective of this study is to extract the graph features from the graph model and then to study the performance of five supervised classification algorithms such as decision tree, random forest, k-nearest neighbour (k-NN), multilayer perceptron (MLP) and support vector machine (SVM) and two unsupervised algorithms such as local outlier factor (LOF) and isolation forest (IF) applied on them. The graph features are extracted using graph algorithms such as degree centrality, label propagation algorithm (LPA) and PageRank algorithm and incorporated with other features in the dataset.

At first, comparison of performances of individual classification models is carried out by applying each graph feature individually and later by combining all the graph features. Thus, the objective of this study is to observe the underlying connection between the transactions, so that they can be represented properly and their properties can be captured in the form of some quantifiable unit by using the above algorithms so as to improve the fraud detection capability of the classification models.

Contribution to this Study

In this study, a fraud detection system has been developed to detect the fraudulent transactions in which, a graph database model is applied for translating the data into a graph database format. Three numbers of graph algorithms such as degree centrality, PageRank and label propagation algorithm (LPA) are used for the extraction of graph features along with other features from the graph database [11]. A good number of researchers have considered the supervised machine learning (ML) techniques for detecting the frauds using credit card. Considering from a pool of large number of supervised and unsupervised ML techniques, five supervised ML techniques such as decision tree, random forest, k-NN, MLP and SVM as well as two unsupervised techniques such as LOF and IF have been considered in this study, because of their improved performances as reported in the literature [12,13]. It is observed that the performance of the models is quite satisfactory on the particular dataset.

The authors have a strong conviction that the study of credit card fraud detection technique based on a graph database model along with various graph-based algorithms and then improving the performance by application of various machine learning techniques is very much a novel one.

The article on credit card fraud detection technique by applying graph database model has been organized in the following manner. In Sect. 2, the literature survey of credit card fraud detection by applying graph database model has been described. Different machine learning techniques are also explained in this section. Various graph algorithms and their applications with classification models are discussed in Sect. 3. The machine learning classification algorithms are discussed in Sect. 4. Section 5 has a complete discussion about the proposed graph model with detailed discussion of the dataset. Implementation and result discussion have been elaborated in Sect. 6. Conclusion and future works are described in Sect. 7.

2 Literature Survey

The literature of various articles on credit card fraud detection using graph database as well as machine learning techniques is extensively explored [14]. Awoyemi et al. have discussed the application of various machine learning techniques to detect credit card fraud [15]. They have considered decision trees [16], random forests [17], k-nearest neighbour, neural networks [18], support vector machines [16], Naïve Bayes [18], logistic regression, anomaly detection, etc., for the classification of fraudulent transactions. Bhattacharyya et al. have provided a list of data mining algorithms used till date for this purpose [19].

Kavitha et al. have mentioned about the usage of graphs and graph features for detection of credit card fraud as well as other financial frauds [7]. Sadowski et al. have mentioned regarding the discover of fraudulent connections using graph databases [8]. Nuno Carneiro et al. have explored the combination of manual and automatic classification of different machine learning methods after observing the complete development process [20]. Eifrem et al. have also discussed one of the aspect of using graph database as a foolproof methodology to detect financial fraud [9].

John et al. have proposed a detailed experimental result of LOF and IF algorithms by using Python to develop a fraud detection model and compared the performances of the algorithms by applying with the dataset [21].

Vijaykumar et al. have considered IF and LOF by preprocessing the data and implemented it. Further, the performance results are analysed by considering the fraud detection system [22].

Hence, it is substantiated that a study on the fraud detection by applying graph database approach called Neo4j database with the incorporation of graph features can be undertaken in order to improve the detection of fraud with certain classification models. The main concern of this research is to observe whether any single graph feature can improve the performance of a classification model.

3 Graph Algorithms used for Classification Models

Several graph algorithms have been implemented by a good number of researchers with classification models to detect the credit card and other financial fraud [8,14].

3.1 Use of Graph Database and Neo4j Tool

In this study, the fraud detection analysis has been carried out based on number of records representing transactions using credit cards and a graph is being developed, where node represents the transaction type and edge represents the relationship between them. Graph-based fraud detection technique has been considered as it helps to identify a fraudulent record. Also, the graph features facilitate to identify the most important nodes and justify the group-dynamics such as credibility, accessibility, the speed at which objects spread and bridges between the group of nodes [8,9].

Hence, three graph algorithms such as Degree centrality, PageRank and LPA have been applied to extract important features in order to improve the classification from the graph generated. Degree centrality and PageRank algorithms are the centrality algorithms, used to emphasize the roles of specific nodes in a graph with their effect on the network [11]. LPA is a community detection algorithm that helps in finding communities and its members will have more relationships within the group than outside their group. Community detection algorithms are often used to produce network visualization and help to extract the necessary graph features from the database to create a graph model.

Neo4j is an open source graph database, developed by using Java technology in 2007 [23,24]. It is highly scalable as well as schema-free (NoSQL) database. Neo4j is accessed by using Cypher Query Language (CQL) through binary bolt protocol or through a transactional HTTP endpoint. In this, each data is stored in the form of a graph node, edge, or attribute. Both nodes and edges in the graph can be

Table 1 Comparison of both relational database and graph database components

Relational database	Graph database
Table	Graph
Row	Node
Columns and data	Properties and its values
Constraint	Relationship
Joins	Traversals
Implementation: MySQL	Implementation: Neo4j

labelled and the labels can be used for narrow searches. Neo4j quickly adapts to the new methods of fraud with faster credit risk analysis. It supports ACID properties of the transactions (Atomicity, Consistency, Isolation and Durability) [25,26]. It has a high-availability clustering for organizational deployment with full transaction support and visual node-link graph explorer. It is accessible from most of the programming languages as it has built-in API interface, with a proprietary Bolt protocol. It deploys graph-based solutions more faster and has easy usability with streamlined workflows.

Graph database is used to model the data in a graphical form by extracting the graph features [27]. Graph databases are used in place of the relational database because the relationship between the nodes is more valuable as compared to the data itself [28]. Unlike relational databases, graph databases store relationships and the connections like first-class entities as shown in Table 1. Several graph databases have been proposed by different researchers for extraction of graph features and Neo4j database has the good performance as of others [29]. The flexible graph model properties of Neo4j graph database ease for different organizations to evolve fraud detection data models, ultimately helps the security teams to match with the pace of ever advancing fraudsters. The Neo4j database stores the data that is structured in graphical manner rather than in tabular form. The advantages of using Neo4j graph database are given below:

- It is a flexible data model and has real-time insights.
- It allows easy retrieval of data through Cypher Query Language.
- It does not use complex methods such as joins

Cypher is a declarative graph query language mainly inspired by SQL, which seeks to avoid the necessity of writing the traversals in coding format. It is used to modify and create graph databases in Neo4j. Neo4j can be used for fraud detection by using graph database. Since there is a strong connection between the ways in which the fraudsters commit crime, it is hereby proposed to depict the previous crimes committed in the form of a graph and use various graph algorithms to find the site where the probability of committing credit card fraud is the maximum. Neo4j tool has several plugins such as Awesome Procedures on Cypher (APOC), Graph Algorithms and GraphQL, which support several graph algorithms.

The graph-based approaches are considered to be prognosticative in nature for detecting the pattern of fraudulent activities by using connected data analysis [14]. Graphs help to provide a practical approach for understanding the links between the financial entities because of proper representation of the transactional information. Since the fraudsters adopt various strategies which are often changing in nature, the emphasis in this study on detection of fraud pattern based on graph theoretic analysis helps to explain possible behavioural patterns, where its features correspond to certain amount of correlations among the fraudulent nodes. The effectiveness of combining the graph features improves the discovery of fraud patterns by applying the connected data analysis. Integrating the graph features optimize the area under precision–recall curve that assists to negotiate between the various rates of true positive and true negative instances.

Popularity of a node measures the relationships between the in-degree and out-degree links of other transactional nodes in the graph [30]. The popular node is that node in the graph, where the prediction of the occurrence of a fraudulent activity is maximum and it is due to the effective utilization of the prior information to find the patterns.

Connectivity of a graph is measured by considering the interrelations among the links and transactional nodes and it helps to discover the connectivity patterns among the higher suspicious nodes [31].

Influence measure of a node quantifies the amount of influence by individual node that helps to maximize the analysis in order to find the fraudulent pattern based on the connectivity [30]. It helps to fix the node sequence according to the descending order of node influence value.

3.1.1 Degree Centrality Algorithm

In a graph, the feature of centrality measures the closeness of a node towards the centre and emphasizes the role of particular nodes with their effects on the network for identifying the anomalous transactional nodes [32]. In the graph database, the queries are explored to find the relationship among the transactional nodes. The graph features are applied to find the structure of the transactional data and discover the fraud patterns. These fraud patterns are useful to find the most predictive frauds by adding them with the classification models. The centrality feature of a node is calculated by taking the mean of the nodes, which are closer towards the centre and the higher mean values indicate the presence of fraudulent nodes.

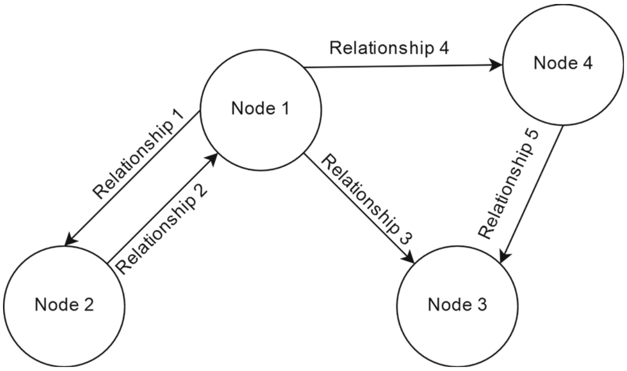

Fig. 1 Relationship among the nodes using degree centrality

This algorithm is used to find the popular nodes in the graph [33]. The popular nodes are the places, where the occurrence of credit card fraud has the maximum probability. It measures the number of incoming and outgoing relationships from a node [34]. Degree Centrality concept was proposed by Linton C. Freeman [33]. Bangcharoensap et al. have further applied this concept in order to separate fraudsters from legitimate users in an online auction [35]. The weighted centrality for fraudsters is significantly higher, because they tend to cooperate in a secret and unlawful manner with each other to increase the price of products artificially. This algorithm has been implemented using Neo4j Graph Algorithms library.

As shown in Fig. 1, the graph algorithm indicates total number of incoming and outgoing relationships from a node. For example, in the following figure, Node 1 has three outgoing relationships and one incoming relationship. Therefore, degree centrality for Node 1 is four. Similarly, the degree centrality for Node 2 is two, for Node 3 is two and for Node 4 also it is two.

3.1.2 PageRank Algorithm

It measures the connectivity and influences the nodes with other neighbours. It is calculated iteratively by distributing a particular node's rank over its neighbours or by randomly traversing throughout the graph nodes and counting the frequency of getting nodes during the traversal [36]. The underlying assumption in this type of algorithm is that the important pages are likely to receive a higher volume of links from other pages. The PageRank algorithm has been used for various application-domains since its inception [37]. This algorithm is used to rank the public places or streets to predict the traffic flow and human movement in those areas. It is also applied to detect fraud in healthcare centres as well as financial institutions [38]. The financial institutions calculate the PageRank value of nodes in transaction graph by applying the graph-based features and various machine learning algorithms are applied to obtain an improved result in fraud detection.

As shown in Fig. 2, initially, Node value = 1/n (Where n = total number of nodes) is considered. In the subsequent passes, the node value is calculated as: Node value = prior in-link values – prior out-link values. For all passes, the link value is calculated as: Link value = Node value / number of out-links.

The algorithm terminates and finds the PageRank values of each node when there is a convergence on a solution or a set of solutions range or a fixed number of iterations. The PageRanks of Node 1, Node 2 and Node 3 are obtained 0.4, 0.4 and 0.2, respectively, at the nth pass as shown in Fig. 2.

3.1.3 Label Propagation Algorithm (LPA)

LPA is a community detection graph algorithm and finds the communities in a graph [39,40]. It uses network structure alone and does not require any other information to find communities [39]. It is a faster algorithm and works by initializing every node with a unique label and propagates these labels through the networks and at every propagation, the label is updated for the node having maximum number of neighbours. Densely connected labels get a unique label. This algorithm has been used to detect tweets on the basis of positive and negative emoticons in combination with the Twitter follower graph. It has also been used to detect the potentially suspicious or normal transaction with the graph nodes [41].

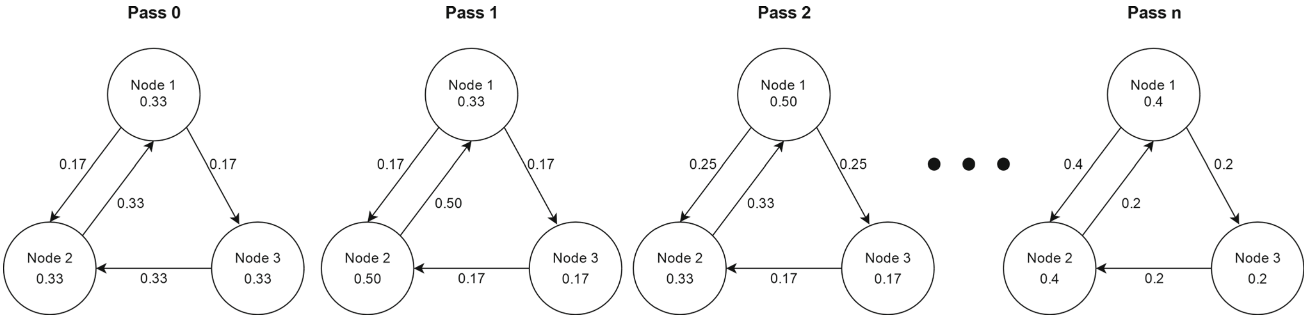

Fig. 2 Graphical representation to find PageRank at different nodes

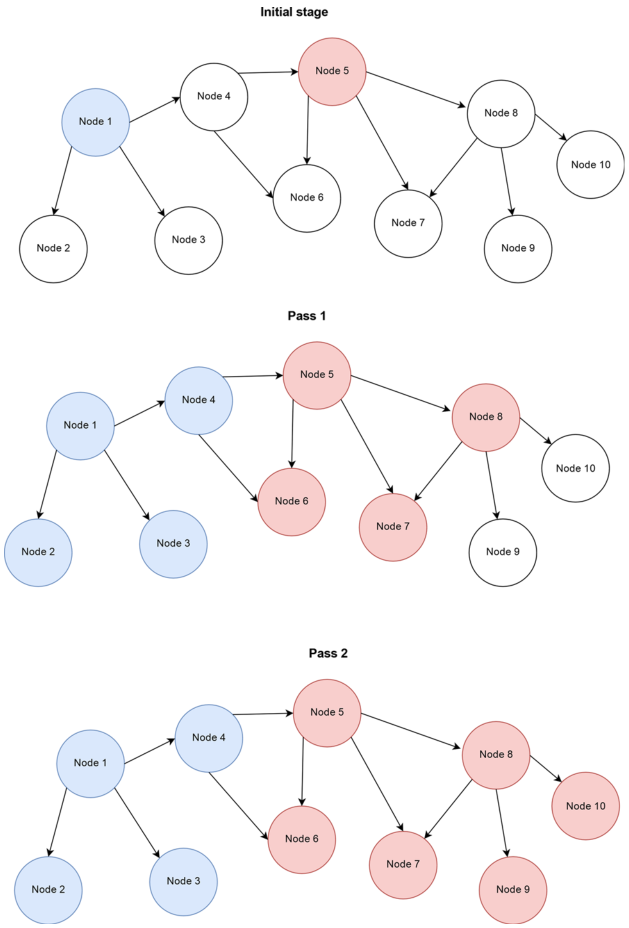

Fig. 3 Graphical representation of nodes using LPA

As shown in Fig. 3, at the initial stage, some nodes have label. In the figure, Node 1 and Node 5 have different labels. In pass 1, the labels are passed to the neighbouring nodes. Thus, Node 2, Node 3 and Node 4 have the same label as Node 1 and Node 6, Node 7 and Node 8 have the same label as Node 5. In the next pass, Node 9 and Node 10 achieve the same label as Node 5. Thus, the algorithm terminates till all the nodes are assigned some label.

4 Machine Learning Classification Models

A good number of researchers and practitioners have proposed various machine learning classification algorithms for the efficient detection of fraudulent transactions in credit card by developing a fraud detection model [42,43]. The classification algorithms with graph database aim to provide the predictive accuracy values with other performance parameters by considering the positive and negative class instances [44]. The fraud detection model responses in the real time to detect the threat accurately in order to curb the loss of the customers as well as financial institutions. The following classification techniques are considered for this research paper.

4.1 Decision Tee Algorithm

This is a machine learning algorithm, mainly used for both classification as well as regression [16]. As shown in Fig. 4, it can be visualized as an upside down tree structure, which spreads its branches so as to predict classes or labels at the leaf nodes. In a decision tree, specific conditions are checked at each branch or decision node to predict its class or label. The decision of which condition is to be checked at what label is made by calculating the information gain or Gini Index of the feature. Gini Split is done by calculating the Gini Index which is given by the difference of ‘1’ and the summation of the squared probabilities of each label. Information gain is obtained by calculating the decrease in entropy of the tree if it is divided on that feature. Gini Index usually favours larger partitions while information gain favours smaller partitions. Thus, a decision tree can also help the observer to understand on what basis the classification is being done. The formulas for calculating the Gini Index, Entropy and Information gain are mentioned below:

$$Gini = 1 - \sum_{i=1}^{c} (p_i)^2 \tag{1}$$

$$Entropy = 1 - \sum_{i=1}^{c} -p_i * log_2 p_i \tag{2}$$

$$InfoGain = entropy(parent) - entropy(children) \tag{3}$$

A simple decision tree for identifying the suspicious or fraudulent transactions is depicted as in Fig. 4. Here, the conditions of the nodes are considered to identify a transaction is fraudulent or not. For the condition *‘transaction amount’*, if it experiences a larger transaction value, then it may be considered as a fraudulent transaction. If not, then the tree checks whether the *‘transaction time’* is unusual. If so, then the chance of fraud cannot be avoided.

4.2 Random Forest Algorithm

Random forest is a supervised machine learning classification algorithm used for classification as well as regression purposes [17,45,46]. According to its name, it is derived from a group or series of decision trees or a forest of trees as shown in Fig. 5. Each individual decision tree in the forest predicts or classifies the class label and the class label with the majority vote from the trees of the random forest is assigned. Thus, it usually gives better performance than a decision tree. The reason is that the trees protect each other from the errors that may have occurred if each of them had individually predicted

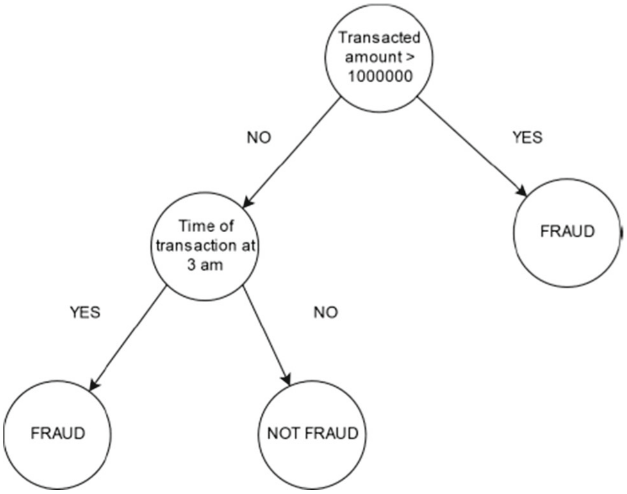

Fig. 4 Simple decision tree to detect fraudulent transactions

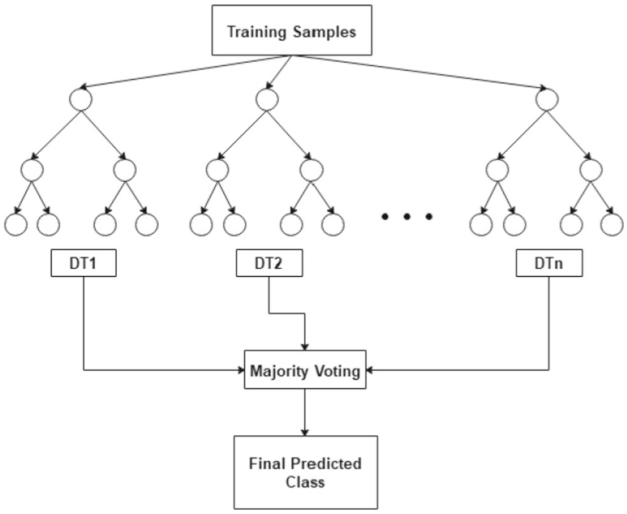

Fig. 5 Block diagram for random forest classification model

the class label. If some of the trees are wrong, others help them to move in the right direction; thus, it helps in rectifying the errors that might have occurred. And for this, there should not be any correlation among the trees.

To ensure that there is minimum correlation among the individual trees, the random forest classifier employs two methods known as Bagging or Bootstrap Aggregation and feature randomness. In Bagging method, the sensitivity of the decision tree towards the kind of data on which they are trained, takes the advantage of individual trees and each tree randomly samples from the dataset with replacement. This results the difference in dataset for each tree and thus less correlation is built. In case of feature randomness, a random forest ensures diversification by allowing the trees to choose a feature to split a node from the subset of features rather taking the best feature as in decision tree.

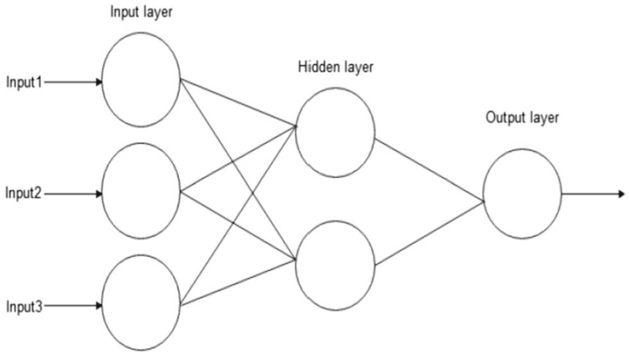

Fig. 6 Simple architecture of an MLP classification model

Random forest has an improved predictive power with a series of decision trees and in turn reduces the variance by training different samples on data. It considers the mean value of the results of the decision trees and that ultimately reduces the risk of overfitting. But if we consider only one tree in the random forest, then it will overfit the data, since it behaves like a single decision tree.

4.3 K-nearest Neighbour (K-NN) Algorithm

K-NN is a supervised machine learning technique, applied for both classification and regression [47,48]. To find the class label of an unknown sample, the algorithm finds its nearest K samples by some distance measurement technique such as Euclidean distance or some parameters. Then, the sample is assigned to the class label to which the majority of its neighbours belong. The difficulty associated with K-NN is that the value of K is difficult to determine. It is not required to build the model before the classification is a benefit of using K-NN. The disadvantages include a non-delivery of straightforward classification probability formula and the proportion of separation and the cardinality K of the area exceptionally influence the predictive accuracy.

4.4 Multilayer Perceptron (MLP) Algorithm

MLP is a kind of feed-forward neural network used for classification as well as regression problems [46,49]. There are minimum three layers in an MLP such as input layer, at least one hidden layer and an output layer as shown in Fig. 6. Apart from the input layer nodes, the other layers have also neurons that utilize an activation function. It uses a supervised learning system known as back-propagation method to find the value of the weights, i.e. the trainable parameters. Its various layers, nonlinear activation functions differentiate Multilayer Perceptron from the linear perceptron. In MLP, all the neurons have a linear activation function, i.e. a linear function that points out the weighted contributions to the output layer.

4.5 Support Vector Machine (SVM) Algorithm

SVM uses statistical learning strategy and has effective application in classification and anomaly detection [16,50]. They are typically related to the neural network systems with kernel functions. This can be seen as an elective strategy to get neural framework classifiers. SVM models are the regulated machine learning methods associated with the variations from the normative acknowledgment (anomaly recognition) in the one-class setting. These strategies use one class learning frameworks for SVM models and take in a region that contains the preparation tests. The basic idea behind the SVM order calculation is to build up a hyperplane as the choice plane by fixing the detachment between both the positive and negative mode expanded.

SVM has two essential properties, i.e. they have bit portrayal and edge enhancement. This model finds an exceptionally phenomenal kind of straight model, i.e. the most outrageous edge hyperplane and it arranges all preparative information examples successfully by confining them through a hyperplane.

In credit card fraud acknowledgment, for each test information occurrence, it always chooses whether the test model falls inside the scholarly region or not. By then, in the event that a test occurrence happens inside the preparation region, at that point it is affirmed true to form generally atypical. This model shows that it has a higher precision of discovery that most of the predicted frauds are correct when contrasted with different empirical calculations of various parameters. It similarly makes some prevalent memories adequacy and speculation limit.

SVM classification method helps in optimizing the boundary value by maximizing the margin of separation and detects the hyperplane which can classify the data instances into the correct classes. SVM algorithm and neural network have different working procedures. Kernel method is an algorithmic class broadly used for pattern analysis with SVM classification algorithm. Kernel function helps in mapping the data from the original space to the higher dimensional space. In neural network, kernels are the set of weights or parameters of the input features [51].

4.6 Local Outlier Factor (LOF)

LOF is an unsupervised machine learning technique, where each sample of data will have some anomaly score [21,22]. In local outlier factor, local deviation is measured with reference to its neighbours. It is based on local density of the samples. The anomaly score of a sample means how isolated that sample is from its neighbouring sample. Local density of a sample can be measured and compared to local densities of its neighbours and thus is used to identify the data samples as outlier with lower density than their neighbours.

4.7 Isolation Forest (IF)

Isolation forest is alike random forest and is built using decision tree but unlike random forest isolation forest identifies anomalies and outliers by selecting the minimum and maximum value of the features [21,22]. It is best suited for anomaly detection that isolates the outliers by randomly selecting a feature from the given set of features and then randomly selecting a split value between the max and min values of that feature. This random partitioning of features will produce shorter paths in trees for the anomalous data points, thus it helps in distinguishing the data instances from the rest of the available data.

5 Proposed Graph Model

In the proposed study, three graph algorithms such as degree centrality, PageRank and label propagation algorithm are considered for extracting the graph features using Neo4j graph database tool. Degree centrality and PageRank algorithms are the centrality algorithms to measure the closeness of any node towards the centre and to emphasize the role of a node in the network for identifying the anomalous transactional nodes. Label propagation algorithm is a community detection algorithm that groups a set of nodes in a graph to identify the pattern for the fraudulent activities. These three graph algorithms have a better impact with the chosen machine learning algorithms with improved accuracy to detect the fraudulent activities.

This study on fraud detection uses the BankSim dataset from Kaggle (http://www.kaggle.com/ntnutestimon/banksim1) [27], which is a preprocessed data, where the features and labels are properly optimized. The preprocessed data includes dropping the features that have only one value. The feature named as amount is scaled to a specific range to control the variance. BankSim dataset is a randomized simulation and is not identical to the original data. Thus, it does not contain any personal information and retains the privacy. Also, the synthetic data is more efficient for testing the model and significantly cost-effective for collecting the real-world data.

The data dictionary provides the complete information for the database with the description of contents, the format of the data elements and their structures. The objective of the BankSim data is to generate a synthetic data that helps to extend the research on fraud detection. In this research work, k-fold cross-validation technique is considered to evaluate the predictive classification models by partitioning the original data sample into a training set and test set to train and test the model, respectively. For training and testing the model, the dataset is split into k consecutive folds (twofold in this case). While kth fold is used for validation, remaining k-1 folds are validated for testing. The entire dataset is divided

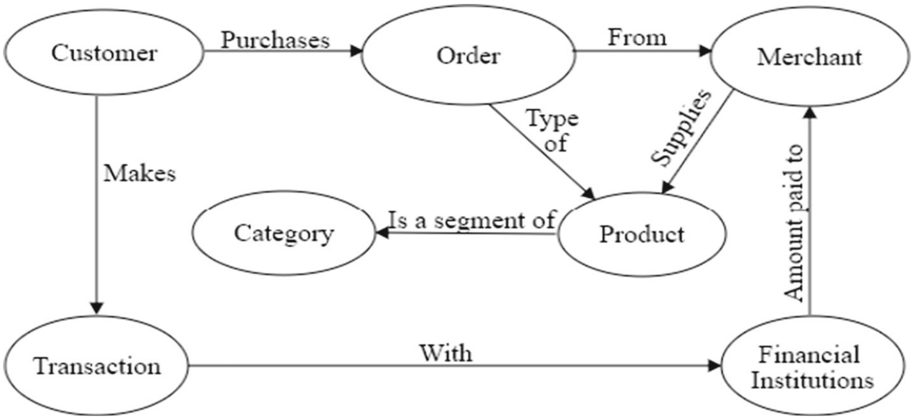

Fig. 7 Meta-graph for the database with nodes and their relationships

into 15 columns such as ‘step’, ‘customer’, ‘age’, ‘gender’, ‘zipcodeOri’, ‘merchant’, ‘zipmerchent’, ‘category’, ‘amount’, ‘fraud’, ‘source’, ‘target’, ‘weight’, ‘typetrans’ and ‘fraud’. It is implemented with 180 steps from 0 to 179. The performance parameters are evaluated by implementing the dataset on the classification algorithms and the results are analysed.

A meta-graph for the database helps to model the data in a virtual graph form with the interconnection between nodes and their relationships. The nodes of the graph represent the entities and the relationship represents the association among the nodes. A graph model is created by merging the desired features and making relationships among them as shown in Fig. 7. Here, a relationship ‘make’ is made from customer to transaction and another relationship ‘with’ is made from transaction to the financial institution to purchase order from merchant.

Preprocessing of the sample data is carried out for scaling purpose of data prior to machine learning algorithms. Finally, all machine learning algorithms are imported from scikit-learn, which will be using for classification method to demonstrate the performance results. In supervised learning, features are the descriptive attributes, and the label is used to predict or forecast. To forecast the predicted result, it uses past data, and taking this data the machine learning classifier uses the attributes as features and the labels are associated with the attributes.

The classification algorithms such as decision tree, random forest, K-NN, MLP, SVM, LOF and IF are implemented first with scikit-learn library by using Python, without considering any graph features of the graph database algorithms and later the classification algorithms are implemented with considering the graph models. It uses *preprocessing.LabelData()* in scikit-learn to process the data, and *train_test_split()* to split the dataset into test and train samples by dividing into k consecutive folds (2 in the case of this experiment). Then, each fold is used once for validation while k-1 remaining folds are validated from the training set. For each of the classification techniques, the performance parameters such as accuracy, precision, sensitivity (recall), F1-score and Matthews correlation coefficient (MCC), receiver operating characteristic curve (ROC) and area under curve (AUC) score, area under the precision–recall curve (AUPR) are empirically evaluated and the values are compared with the machine learning classification algorithms. It is observed that LOF is a more improved method and provides higher recall value as compared to other methods.

The next step is used to create graph model with the help of Neo4j and Cypher Query Language (CQL) [52]. Graph database is used for modelling the data in the form of a graph where the nodes of the graph are depicted as entities while the relationship is depicted as association of these nodes [53]. Unlike the relational databases, graph databases store relationships and their connections as first-class entities. Constraints are being created on the ‘Customer’ and ‘Bank’ nodes so that they will be unique. The dataset is used to detect fraudulent transactions with Neo4j tool using cypher queries.

A graph model is created by merging the desired features and making relationships among them as shown in Fig. 8. Here, a relationship ‘make’ is made from ‘N’ number of customers to the transactions they do and another relationship ‘with’ is made from ‘N’ number of transactions to the financial institutions, where the transaction is carried out.

In the financial transaction process, ‘n’ number of payers or customers do their transactions in which an n-partite graph or multipartite graph is being considered. Graph algorithms implemented in Neo4j tool need a multipartite graph model for its operation. Therefore, a multipartite representation is created as shown in Fig. 9 by making customers and merchants of the same label, say Payer. Further, the degree centrality, PageRank and label propagation algorithms are applied on this graph using Neo4j tool and APOC library, which are enabled in its plugins.

The graph features are extracted by using the algorithms such as Degree centrality, PageRank and LPA from Neo4j tool along with Py2neo library and incorporated with other

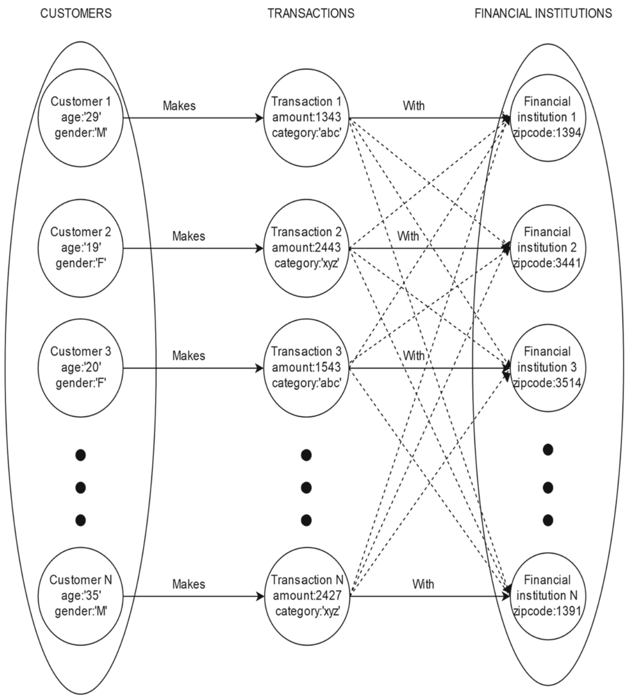
CUSTOMERS
TRANSACTIONS
FINANCIAL INSTITUTIONS
Customer 1 age:'29' gender:'M'
Customer 2 age:'19' gender:'F'
Customer 3 age:'20' gender:'F'
Customer N age:'35' gender:'M'
Makes
Transaction 1 amount:1343 category:'abc'
Transaction 2 amount:2443 category:'xyz'
Transaction 3 amount:1543 category:'abc'
Transaction N amount:2427 category:'xyz'
With
Financial institution 1 zipcode:1394
Financial institution 2 zipcode:3441
Financial institution 3 zipcode:3514
Financial institution N zipcode:1391

Fig. 8 Graphical representation of the transactions with the financial institutions

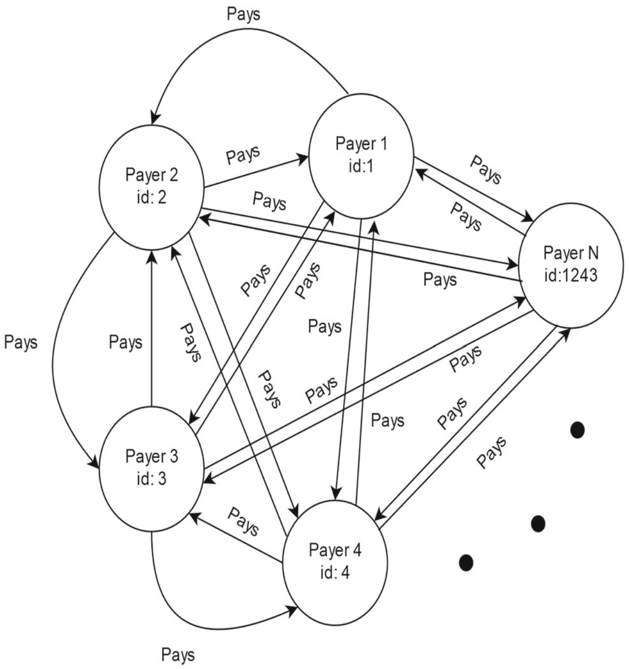

Fig. 9 Multipartite graph representation for the transaction between the payers

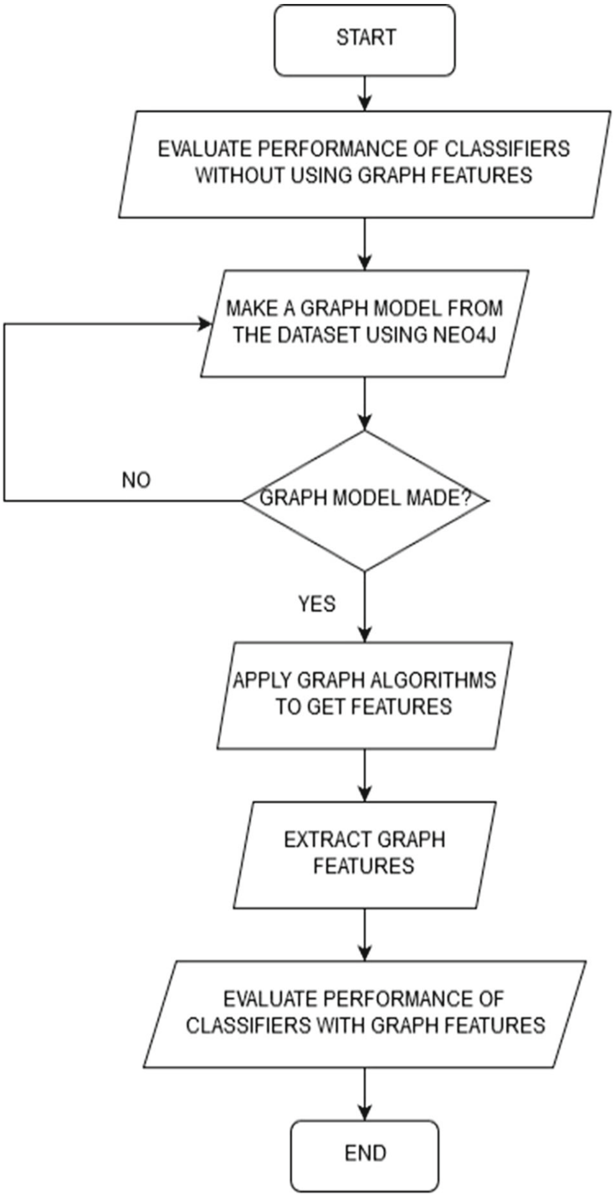
START
EVALUATE PERFORMANCE OF CLASSIFIERS WITHOUT USING GRAPH FEATURES
MAKE A GRAPH MODEL FROM THE DATASET USING NEO4J
GRAPH MODEL MADE?
NO
YES
APPLY GRAPH ALGORITHMS TO GET FEATURES
EXTRACT GRAPH FEATURES
EVALUATE PERFORMANCE OF CLASSIFIERS WITH GRAPH FEATURES
END

Fig. 10 Flowchart showing methodology using graph model

features. Py2neo is a library from the client side having the toolkit that helps to work with Neo4j from Python applications as well as from the command line. The package in Py2neo database contains both classes and functions needed to interact with a Neo4j server. The graph class that represents the Neo4j graph database instances provides access to a large part of Py2neo API. Finally, all the classification models mentioned earlier are run on this new dataset and the improvement is noted. A flowchart of this algorithm is presented in Fig. 10.

Steps for Algorithm Implementation of the Graph Model

Input BankSim dataset

Step 1 Evaluate the performance of machine learning classifiers (Such as Decision tree, random forest, k-NN, MLP, SVM, LOF and IF) by considering the dataset.

Step 2 While graph model is not created do

Step 2.1 Make graph model from dataset using Neo4j tool. End while

Step 3 Apply graph algorithms (Degree centrality, LPA and PageRank) one by one.

Step 4 Extract the features obtained by applying graph algorithms.

Step 5 Evaluate performance of each classifier using graph features.

Output Comparison of classifiers with and without using graph features.

6 Implementation and Result

A graph database based on the sample data is developed using Neo4j tool, where Cypher Query Language (CQL) is used to access the database from the software written in a language either by a transactional HTTP endpoint or by the binary bolt protocol. In this, data are stored in node, edge or attribute form. Cypher is a declarative graph query language used to modify and create graph databases in Neo4j tool.

6.1 Dataset used for Implementation

A synthetic dataset from a financial payment system has been considered for analysis in this study [27]. This is a synthetic dataset generated by the BankSim payment simulator and available on Kaggle. BankSim was run for 180 steps (approx. for six months); several times and the parameters were calibrated in order to obtain a distribution that is close enough to be reliable for testing and fraudulent transactions were injected into it. Thus, it has 594643 records in total, out of which 587443 (98.79%) transactions are normal payments and 7200 (1.21%) are fraudulent transactions. Since this is a randomized simulation, the values are not identical to the original data. Thus, this dataset does not contain any personal information regarding the customer transactions.

6.2 Performance Parameters

The parameters are evaluated by using the confusion matrix, which has four components such as 'true positive', 'false positive', 'true negative' and 'false negative'. A binary classifier is a classification method that predicts two class labels and thus gives either positive or negative class. Based on this, the output can either be one of the following four types.

- **TP** = True positive (the number of samples which are positive and have been classified as positive by using the classifier). In this case, when a transaction is predicted as fraud and actually it is a fraudulent transaction.
- **FP** = False positive (the number of samples that are actually negative but have been classified as positive by using the classifier). In this case, a transaction is predicted as fraud but actually it is a non-fraudulent transaction. It increases the misclassification of the data. False positive should be minimized to improve the accuracy since it predicts a normal transaction as fraud.
- **TN** = True negative (the number of samples which are negative and have been classified as positive by using the classifier). In this case, a transaction is predicted as not fraudulent and actually it is a non- fraudulent transaction.
- **FN** = False negative (the number of samples which are actually positive but have been classified as negative by using the classifier). In this case, a transaction is predicted as non-fraud but actually it is a fraudulent transaction. It also leads to misclassification of data.

6.2.1 Accuracy

Accuracy is an evaluation metric that is used to determine how many samples have been correctly classified from the total samples. It is given by dividing the total number of correct samples with the total number of samples. It gets a value between 0 and 1 and the best accuracy value is 1.0. It is expressed mathematically by the following equation.

$$Accuracy = (TP + TN)/(TP + TN + FP + FN). \tag{4}$$

6.2.2 Precision

Precision is an evaluation metric used to determine how many positive samples classified by the classifier are actually positive. It exhibits how much precise the classifier is. It is given by dividing the total number of correctly classified positive samples by the total number of positive samples classified by the classification model. It is also known as positive predictive value (PPV). It gets a value between 0 and 1 and the best value is 1.0. It is expressed mathematically by considering the following equation.

$$Precision = TP/(TP + FP). \tag{5}$$

6.2.3 Recall or Sensitivity

Recall is an evaluation metric that is used to determine actually how many positive samples are classified as positive by using the classification model. It is also known as true positive rate (TPR) or sensitivity. It is given by dividing the total number of correctly classified positive samples by the total number of actual positive classes. It gets a value between 0 and 1 and the best value of recall is 1.0. It is expressed mathematically by the following equation.

$$Recall = TP/(TP + FN). \tag{6}$$

6.2.4 F1-Score

F1-score is an evaluation metric which takes into consideration of both accuracy and precision. It is given by the value of the harmonic mean of accuracy and precision. It measures the model's testing accuracy on the dataset. It is also known as F-measure. It gets a value between 0 and 1 and the best value is 1.0. It is expressed mathematically by the following formulae.

$$F1{-}score = 2TP/(2TP + FP + FN). \tag{7}$$

6.2.5 Matthews Correlation Coefficient (MCC)

It is an evaluation metric which is used to measure the quality of binary classification. It is generally used when the number of positive and negative samples is imbalanced. It returns the correlation index number between -1 and +1. '-1' means that the samples are negatively correlated while '+1' means that they are perfected correlated. It is expressed mathematically by the following equation.

$$\text{MCC} = \frac{(TP * TN - FP * FN)}{\sqrt{(TP + FP) * (TP + FN) * (TN + FP) * (TN + FN)}}. \tag{8}$$

6.2.6 ROC–AUC Score

ROC (receiver operating characteristic) and AUC (area under curve) score is an evaluation metric, which is mainly calculated from the ROC curve. The ROC curve is plotted by taking the values of sensitivity or recall or (true positive rate) TPR on the y-axis and the values of '1-specificity' or false positive rate (FPR) on the x-axis where TPR and FPR are provided by the respective formula as given below.

$$TPR = TP/(TP + FN) \tag{9}$$

$$FPR = FP/(TN + FP). \tag{10}$$

As the value of sensitivity increases, the value of specificity decreases and vice versa. The area under the curve (AUC) gives the better AUC–ROC score. Its value varies in between 0 and 1. The value 0.0 indicates that the samples are reversely predicted. The value 0.5 indicates that most of the samples are randomly predicted and any score more than 0.5 indicates a better prediction. The AUC–ROC score indicates how well the probabilities from the positive classes are separated from the negative classes.

6.2.7 AUPR Measure

The area under the precision–recall curve is the average of precision across all recall values. AUPR curve is very much useful in the machine learning area particularly for imbalanced data since it focuses on the small data values. It measures the performance with precision and recall. A fraud detection model can achieve the perfect AUPRC, when it can find all the fraudulent transactions without marking the normal transactions as fraudulent one.

6.3 Implementation of Graph Algorithms

Five supervised and two unsupervised machine learning classification models are tested by using BankSim dataset. Theft card, cloned card and internet purchase transactions are associated with BankSim dataset and various types of fraudulent transactions such as 'travel', 'transportation', 'health', 'hotel service', 'sport and toy' and 'fashion' are present with them. The fraudulent activities of these transactions are detected by using the graph properties with the help of graph database model. Firstly, the models are implemented without the inclusion of graph features of the graph database algorithms as shown in Fig. 10. The performance parameters of all the models are noted. Then, a graph database model has been developed by extracting the graph features from the dataset with the help of Neo4j tool. Once the graph model is developed, three graph algorithms such as Degree centrality, PageRank and LPA are applied to extract the graph features from the dataset. Finally, the performance parameters are evaluated for the models with the inclusion of graph features. Linear data structure such as array is used in the process when graph features of the graph database model are not applied.

Total number of instances in the working dataset (http://www.kaggle.com/ntnu-testimon/banksim1) [27] has 594643, of which 587443 are normal transactions and only 7200 are fraudulent transactions. In a credit card fraud detection system, the financial institutions intend to identify and take necessary corrective actions for every fraudulent transaction. Only 1.21% of the transactions are fraudulent, which is very less as compared to the total instances present in the dataset. Among all the machine learning classification algorithms, LOF has the highest accuracy of 99.753% and recall value of 87.791% when all three graph features are considered. The graph features have the significant roles in increasing the evaluation metrics of the classification algorithms. The average percentage value of recall as shown in Table 6 is improved while considering all the three graph features.

6.4 Result Discussion

The graph algorithms are applied with the machine learning classification models and their performances are discussed. Degree centrality is used to classify the fraudsters (internet fraud) from legitimate users during online transactions. The centrality of the fraudsters significantly increases since they do the abnormal activities for doing the transactions. From Table 2, it is observed that with the inclusion of degree centrality feature, there is a significant improvement in the performance metrics of decision tree algorithm though there is a bit improvement in all the classification techniques. By comparing all values, it is observed that LOF yields a better result for this dataset. The evaluation metrics for each model are compared using 2D plots as shown in Fig. 11. The accuracy values of all the classifiers are too close to each other. LOF has 99.541% accuracy value and 83.397% recall value with the degree centrality graph feature.

PageRank algorithm is useful for detecting the behavioural fraud, where the fraudster performs his transactions in an unusual manner. By using the graphical method, the machine learning classification models are implemented to classify the fraud. In Table 3, by the inclusion of PageRank graph feature, it is observed that there is an improvement in all the classifiers except random forest. From the results, it may be observed that k-nearest neighbour (K-NN) algorithm and isolation forest (IF) yield an improved result on classification in such case. The evaluation metrics for each classifier are compared using 2D plots as shown in Fig. 12. It is observed that there is a significant increase in the performance values with PageRank graph feature as compared to without PageRank graph feature. The accuracy value of random forest is 99.478% and recall value of LOF is 84.007% while considering the PageRank graph feature.

LPA is used to infer the features for a machine learning model to track the fraudsters intention with the knowledge of graph features and their relations. As shown in Table 4, the inclusion of LPA graph feature shows an improvement in all the classification models, except precision for random forest. In this case, MLP, LOF and IF yield a better result for classification. The evaluation metrics for each classifier are compared using 2D plots as shown in Fig. 13. It has been observed that the performance values are significantly improved with LPA as compared to without LPA. The accuracy percentage of LOF is found to be 99.483% and recall value percentage is 76.660% while considering the LPA graph feature.

The performance parameters of the combined graph feature model, where the features of all the three graph models are hybridized, are shown in Table 5. In this case, among all the classification models local outlier factor (LOF) algorithm gives the best classification accuracy value as well as recall value as compared to other classification models. With the inclusion of graph features the fraud detection model efficiently analyses the unusual patterns in the financial transactional data to identify the fraud and improves the accuracy as well as other performance with minimized false alarm as compared to without graph features.

The evaluation parameters of each machine learning classification technique are empirically estimated with and without considering graph features of the graph database by using three graph algorithms. With the inclusion of graph features, the performance results of all the parameters are significantly improved as compared to those obtained without the use of graph features. The combined performance of all the three graph algorithms with the BankSim dataset upon the classification techniques has significant improvement in the result as shown in Table 5. In this category, LOF has highest accuracy value of 99.753% and highest recall value of 87.791%. The improvement of average percentage of accuracy, recall (sensitivity) and other performance metrics are exhibited in Table 6 with the consideration of all the three graph features.

The incorporation of all the three graph features improves the classification capabilities of each model to a certain extent. Thus, it may be concluded that graph features help to improve the fraud detection technique by implementing these classification models. The evaluation metrics for each classifier are compared using 2D plots as shown in Fig. 14. It is observed that, there is an improvement in the performance parameters by considering all three graph features as compared to without graph features. The accuracy percentage of LOF has been significantly improved with incorporation of graph features as compared to other techniques. Also, the recall value is observed to be highest for LOF as compared to others.

With the available voluminous transactional data of 594643, only 7200 number of transactions are found to be fraudulent, i.e. 1.21% of the total transactions. However, it is intended to detect every single financial fraud associated with the transactional data and take necessary corrective action with the help of fraud detection model. The improved accuracy percentage as well as the recall value for various

Table 2 Comparison of performance parameters of various machine learning algorithms with and without using degree centrality graph feature in percentage

Classification model	Degree centrality	Accuracy	Precision	Recall	F1-score	MCC	ROC–AUC score	AUPR measure
Decision tree	Without degree centrality	99.150	65.854	76.416	70.743	70.120	87.922	68.231
Decision tree	With degree centrality	99.347	72.480	77.701	75.001	74.523	88.654	76.334
Random forest	Without degree centrality	98.463	82.781	71.897	76.956	76.779	85.848	70.232
Random forest	With degree centrality	99.473	83.867	70.682	76.712	76.648	85.253	79.771
K-NN	Without degree centrality	98.789	76.191	64.263	69.721	70.988	81.243	68.007
K-NN	With degree centrality	99.330	77.859	66.486	71.724	71.161	83.105	73.223
MLP	Without degree centrality	98.789	85.623	54.722	66.771	65.215	77.119	71.435
MLP	With degree centrality	99.346	84.824	56.452	67.789	68.380	78.156	75.090
SVM	Without degree centrality	98.192	77.452	51.123	61.592	62.822	78.193	74.657
SVM	With degree centrality	99.389	79.122	51.995	62.752	65.918	80.191	80.990
LOF	Without degree centrality	98.189	78.102	81.495	79.762	66.318	82.229	78.228
LOF	With degree centrality	99.541	79.113	83.397	81.199	64.908	79.199	87.380
IF	Without degree centrality	99.002	80.225	80.992	80.607	71.002	80.141	77.337
IF	With degree centrality	99.381	81.162	82.997	82.069	69.829	80.071	84.994

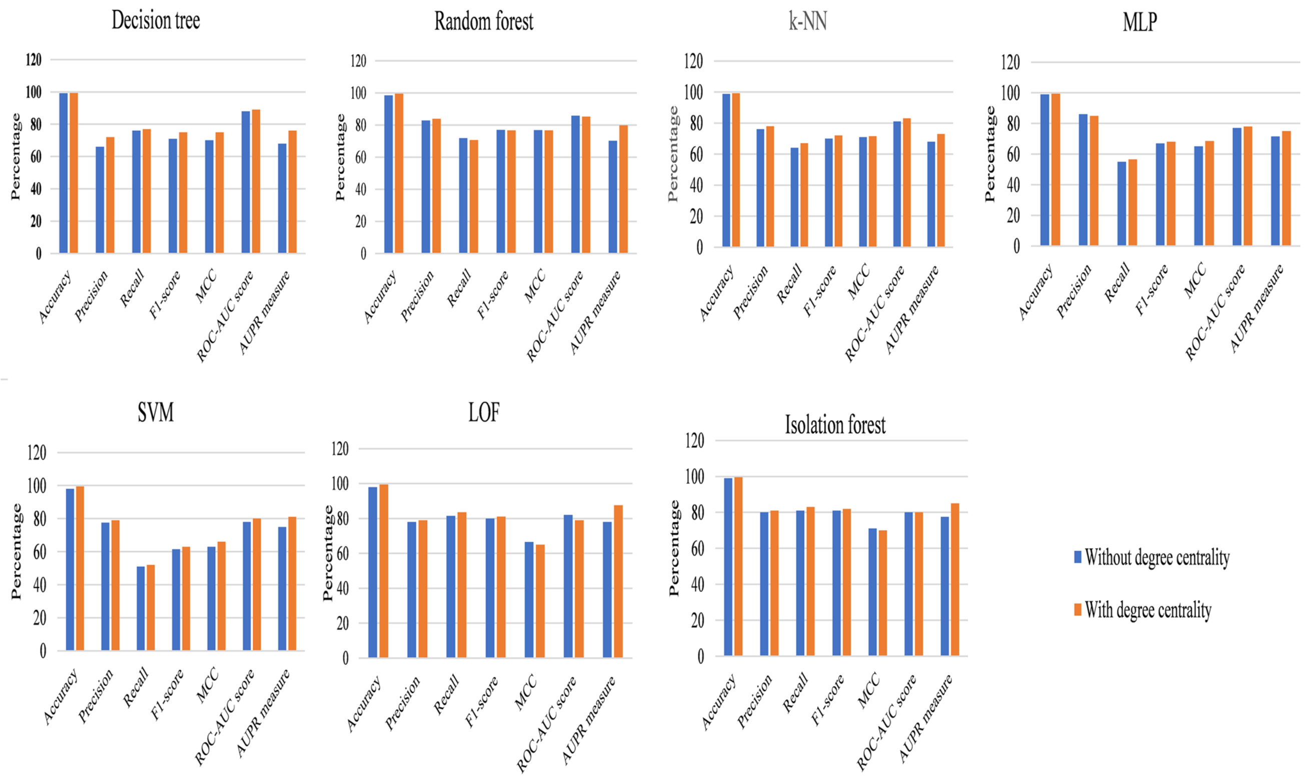

Fig. 11 Comparative analysis of performance metrics of different machine learning classification models with and without consideration of degree centrality graph features

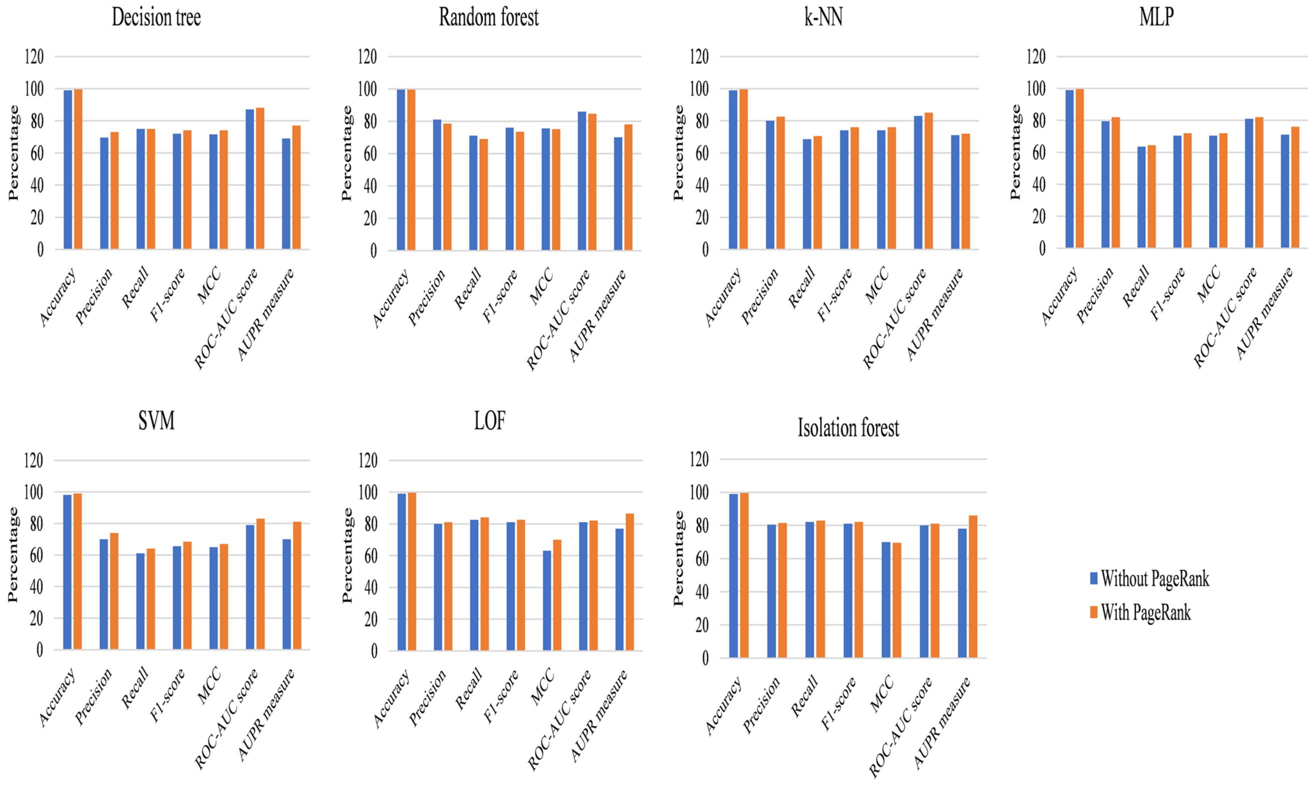

Fig. 12 Comparative analysis of performance metrics of different machine learning classification models with and without consideration of PageRank algorithm graph features

Table 3 Comparison of performance parameters of various machine learning algorithms with and without using PageRank graph feature in percentage

Classification model	PageRank	Accuracy	Precision	Recall	F1-score	MCC	ROC–AUC score	AUPR measure
Decision tree	Without PageRank	99.237	69.561	74.934	72.148	71.482	87.233	69.093
Decision tree	With PageRank	99.328	72.883	75.732	74.280	73.692	87.672	77.194
Random forest	Without PageRank	99.430	81.109	71.066	75.756	75.523	85.421	69.936
Random forest	With PageRank	99.478	78.573	68.884	73.410	75.111	84.316	77.711
K-NN	Without PageRank	98.790	80.168	68.388	73.811	74.218	83.020	71.199
K-NN	With PageRank	99.440	82.453	70.340	75.916	75.668	85.065	72.109
MLP	Without PageRank	98.790	79.427	63.493	70.572	70.398	80.914	70.991
MLP	With PageRank	99.396	81.912	64.441	72.134	72.332	82.131	76.192
SVM	Without PageRank	98.122	70.131	61.294	65.415	65.291	78.992	69.706
SVM	With PageRank	99.021	73.912	63.992	68.595	66.922	83.189	81.117
LOF	Without PageRank	99.009	79.771	82.455	81.091	63.224	81.002	77.118
LOF	With PageRank	99.318	80.812	84.007	82.379	69.991	81.909	86.310
IF	Without PageRank	98.702	80.209	82.192	81.188	69.772	80.196	78.003
IF	With PageRank	99.282	81.328	83.107	82.208	69.228	80.995	86.076

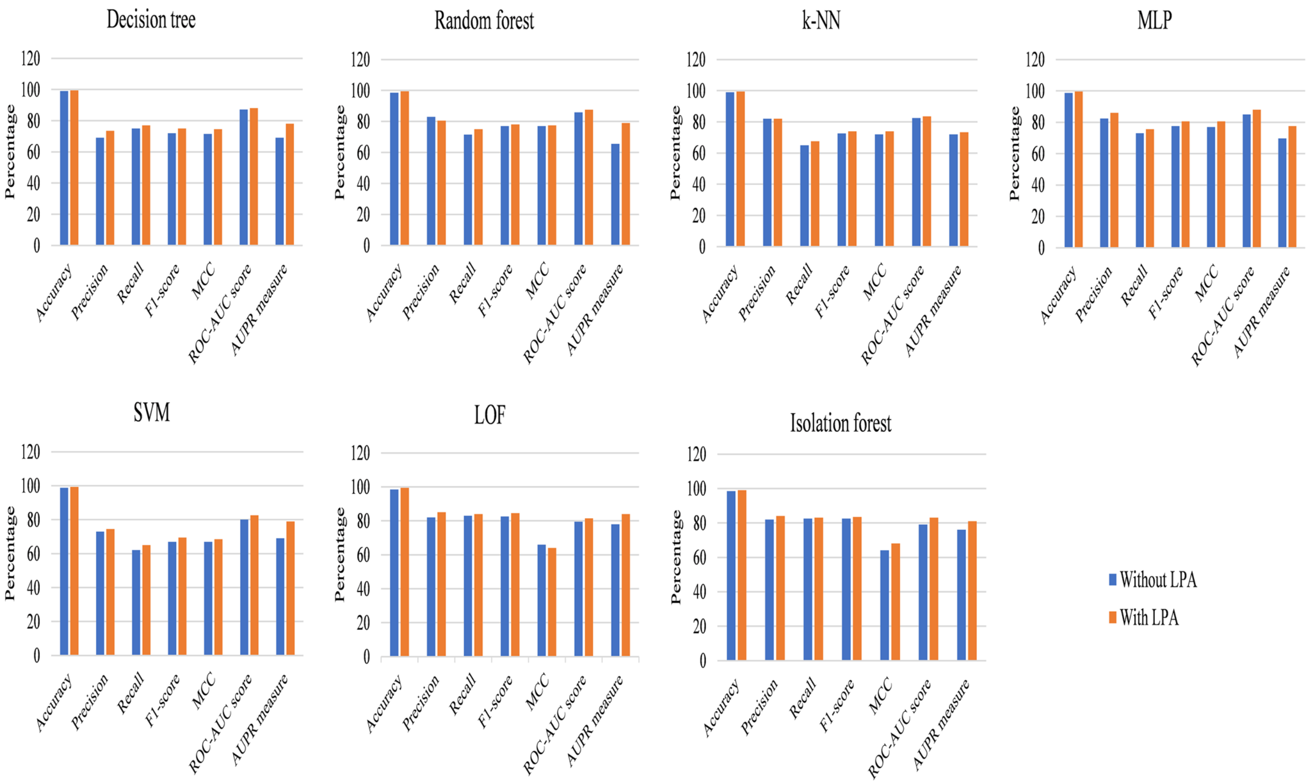

Fig. 13 Comparative analysis of performance metrics of different machine learning classification models with and without consideration of LPA graph features

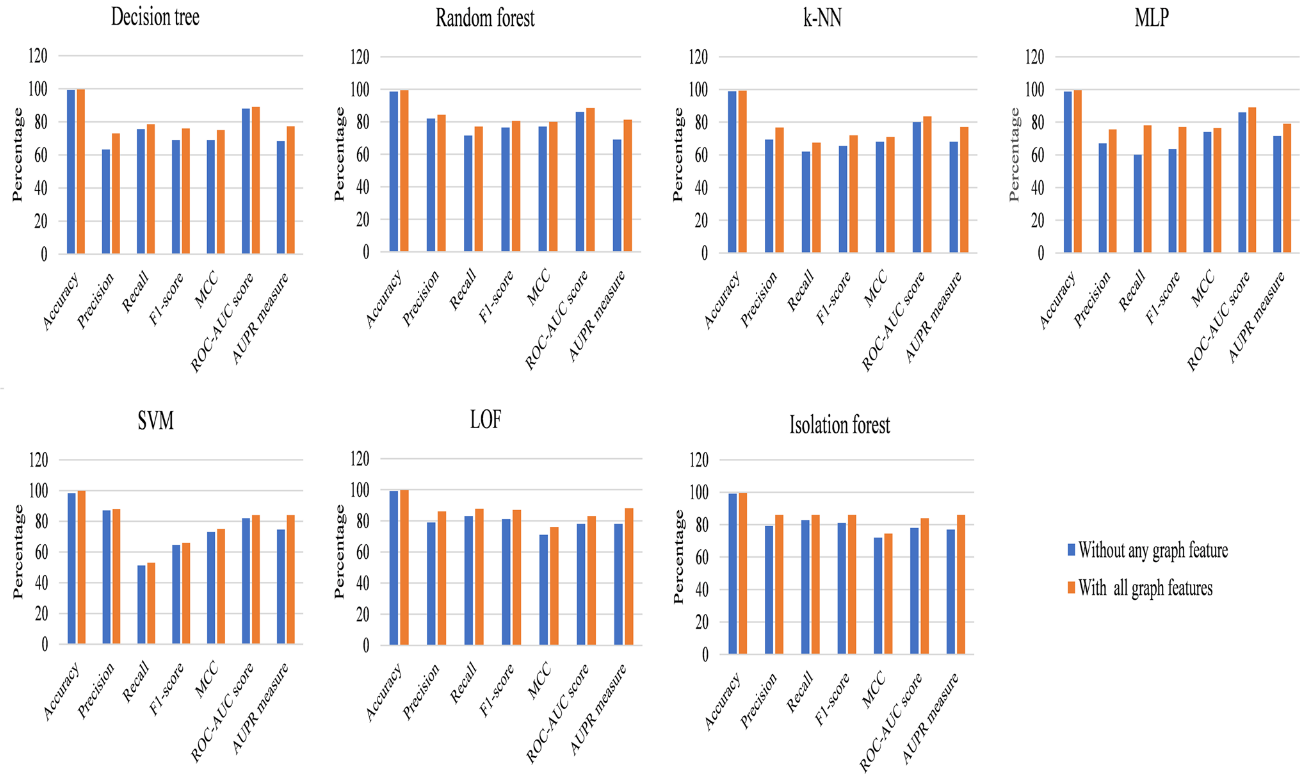

Fig. 14 Comparative analysis of performance metrics of different machine learning classification models with and without consideration of all three graph features

Table 4 Comparison of performance parameters of various machine learning algorithms with and without using LPA graph feature in percentage

Classification model	LPA	Accuracy	Precision	Recall	F1-score	MCC	ROC–AUC score	AUPR measure
Decision tree	Without LPA	99.235	69.069	74.877	71.856	71.234	87.204	68.760
Decision tree	With LPA	99.352	73.393	76.660	74.991	74.479	88.143	78.220
Random forest	Without LPA	98.470	83.212	71.424	76.869	76.786	85.618	65.546
Random forest	With LPA	99.459	80.412	75.188	77.712	77.322	87.470	79.110
K-NN	Without LPA	98.791	81.980	64.687	72.314	72.165	82.440	71.788
K-NN	With LPA	99.411	82.085	67.412	74.028	73.896	83.605	73.259
MLP	Without LPA	98.690	82.298	73.125	77.441	77.246	85.218	69.665
MLP	With LPA	99.449	86.049	75.489	80.424	80.309	87.669	77.447
SVM	Without LPA	98.790	72.783	62.183	67.067	67.246	80.184	69.006
SVM	With LPA	99.259	74.346	65.197	69.472	68.642	82.432	79.110
LOF	Without LPA	98.498	81.992	83.087	82.536	66.190	79.334	78.175
LOF	With LPA	99.483	84.802	84.117	84.458	63.786	81.443	84.184
IF	Without LPA	98.502	82.109	82.634	82.371	64.112	78.661	75.881
IF	With LPA	99.152	83.709	83.125	83.416	68.111	82.904	81.011

Table 5 Comparison of performance parameters of various machine learning algorithms with and without using all three graph features in percentage

Classification model	All graph features	Accuracy	Precision	Recall	F1-score	MCC	ROC–AUC score	AUPR measure
Decision tree	Without graph features	99.365	63.333	75.727	68.978	68.778	88.038	68.232
Decision tree	With graph features	99.477	73.097	78.545	75.723	75.253	89.081	77.332
Random forest	Without graph features	98.451	82.025	71.472	76.386	76.850	86.250	69.119
Random forest	With graph features	99.534	84.311	77.222	80.611	80.286	88.512	81.283
K-NN	Without graph features	98.789	69.266	62.121	65.500	68.136	80.012	68.009
K-NN	With graph features	99.326	76.689	67.538	71.823	71.196	83.623	77.198
MLP	Without graph features	98.672	67.184	60.178	63.488	74.138	85.925	71.440
MLP	With graph features	99.423	75.424	77.873	76.629	76.286	88.782	79.196
SVM	Without graph features	98.192	87.192	51.222	64.533	72.942	82.128	74.658
SVM	With graph features	99.789	87.922	53.134	66.238	75.218	84.129	83.997
LOF	Without graph features	99.184	78.991	83.192	81.037	71.022	78.190	78.238
LOF	With graph features	**99.753**	86.210	**87.791**	86.993	75.991	83.220	88.007
IF	Without graph features	99.070	79.205	82.771	80.948	71.776	78.213	77.237
IF	With graph features	99.539	86.003	86.021	86.012	74.390	84.128	85.774

classification models indicate the capability of identifying the fraudulent activity associated with a financial transaction.

Table 6 represents the average percentage change in different performance metrics when the graph algorithms are considered. The percentage change for three independent graph algorithms and also the percentage improvement for the combined graph features are shown in Table 6. The percentage improvement for recall value is observed to be highest among all others when all the three graph features are considered.

7 Conclusion and Future Work

Thus, it may be considered that the inclusion of graph features from the graph algorithms helps to improve the performance of the machine learning algorithms. In this paper, the performance parameters like accuracy, F1-score, precision, recall, MCC, ROC–AUC score and AUPR measure are evaluated and compared by considering five supervised machine learning classification algorithms such as decision tree, random forest, K-NN, MLP and SVM and two unsupervised machine

Table 6 Average percentage improvement in the evaluation metrics with inclusion of graph features

Improvement in %	Degree centrality (Table 2)	PageRank (Table 3)	LPA (Table 4)	All graph features (Table 5)
Accuracy	0.784	0.440	0.390	0.536
Precision	2.052	1.862	1.392	5.610
Recall	0.978	0.844	2.732	6.718
F1-score	1.658	1.018	2.686	3.020
MCC	2.142	0.962	1.994	3.480
ROC–AUC score	1.008	1.362	1.730	2.354
AUPR measure	1.698	0.992	1.769	2.873

learning algorithms such as LOF and isolation forest. In each case, the performance of the classification models is found to be improved with the incorporation of the graph features extracted from the graph database. The graph features included in this paper have shown significantly improved result as compared to other existing graph features.

The graph features have the significant roles in increasing the evaluation metrics of the classification algorithms. With the large volume of data in the dataset, a smaller number of fraud instances are present, which is very difficult to detect and it is intended to identify same. LOF algorithm has the improved accuracy result and highest recall value with all the graph features.

Other graph algorithms such as closeness centrality, betweenness centrality, Louvain modularity, node clustering coefficient and average clustering coefficient may be considered in future to extract features and applying deep learning algorithms to classify mere critically, the transactions into fraudulent and authentic transactions.

References

1. Zojaji, Z., Atani, RE., Monadjemi, AH.: Survey of credit card fraud detection techniques: data and technique oriented perspective." arXiv preprint arXiv:1611.06439, (2016)
2. Abdallah, A.; Maarof, M.A.; Zainal, A.: Fraud detection system: a survey. J. Netw. Computer Appl. **68**, 90–113 (2016)
3. Lebichot, B., Braun, F., Caelen, O., Saerens, M.: "A graph-based, semi-supervised, credit card fraud detection system."*In: International Workshop on Complex Networks and their Applications*, pp. 721–733. Springer, Cham, (2016)
4. Bhatla, T.P.; Prabhu, V.; Dua, A.: Understanding credit card frauds. Cards Bus. Rev. **1**(6), 1–15 (2003)
5. Hussein, K.W.; Sani, N.F.M.; Mahmod, R.; Abdullah, M.T.K.: Enhance Luhn algorithm for validation of credit cards numbers. Int. J. Comput. Sci. Mob. Comput **2**(7), 262–272 (2013)
6. Laleh, N., Azgomi, MA.: "A taxonomy of frauds and fraud detection techniques." *In: International Conference on Information Systems, Technology and Management*, pp. 256-267. Springer, Berlin, Heidelberg, (2009)
7. Kavitha, M.; Suriakala, M.: Fraud detection in current scenario, sophistications and directions: a comprehensive survey. Int. J. Computer Appl. **975**, 8887 (2015)
8. Sadowski, Gorka.; Rathle, Philip.: Fraud detection: discovering connections with graph databases. In: White Paper-Neo Technology-Graphs Everywhere, pp. 1–10 (2014)
9. Eifrem, E.: Graph databases: the key to foolproof fraud detection. Computer Fraud Secur. **2016**, 5–8 (2016)
10. Schindler, Timo.: "Anomaly detection in log data using graph databases and machine learning to defend advanced persistent threats." *arXiv preprint*, arXiv:1802.00259, (2018)
11. Needham, M.; Hodler, A.E.: Graph Algorithms: Practical Examples in Apache Spark and Neo4j. O'Reilly Media, Newton (2019)
12. Varmedja, D., Karanovic, M., Sladojevic, S., Arsenovic, M., Anderla, A.: "Credit card fraud detection-machine learning methods." *In: 2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH)*, pp. 1–5. IEEE, (2019)
13. Jain, Y.; Namratatiwari, S.D.; Jain, S.: A comparative analysis of various credit card fraud detection techniques. Int. J. Recent Technol. Eng. **7**(5S2), 402–407 (2019)
14. Akoglu, Leman; Tong, Hanghang; Koutra, Danai: Graph based anomaly detection and description: a survey. Data Min. Knowl. Discov. **29**(3), 626–688 (2015)
15. Awoyemi, JO., Adetunmbi, AO., Oluwadare, SA.: "Credit card fraud detection using machine learning techniques: a comparative analysis." *In: 2017 International Conference on Computing Networking and Informatics (ICCNI)*, pp. 1–9. IEEE, (2017)
16. Sahin, Yusuf G., Duman, Ekrem.: "Detecting credit card fraud by decision trees and support vector machines." *In: Proceedings of the International MultiConference of Engineers and Computer Scientists*, Vol I, (2011)
17. Xuan, Shiyang., Liu, Guanjun., Li, Zhenchuan., ShuoWang, Lutao Zheng., Jiang, Changjun.: "Random forest for credit card fraud detection." *In: 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC)*, pp. 1-6. IEEE, (2018)
18. Maes, S., Tuyls, K., Vanschoenwinkel, B., Manderick, B.: "Credit card fraud detection using Bayesian and neural networks." *In: Proceedings of the 1st International Naiso Congress on Neuro Fuzzy Technologies*, pp. 261-270. (2002)
19. Bhattacharyya, S.; Jha, S.; Tharakunnel, K.; Westland, J.C.: Data mining for credit card fraud: a comparative study. Decis. Support Syst. **50**(3), 602–613 (2011)
20. Carneiro, Nuno; Figueira, Goncalo; Costa, Miguel: A data mining based system for credit-card fraud detection in e-tail. Decis. Support Syst. **95**, 91–101 (2017)
21. John, Hyder; Naaz, Sameena: Credit card fraud detection using local outlier factor and isolation forest. Int. J. Comput. Sci. Eng. **7**, 1060–1064 (2019)
22. Vijayakumar, V.; Sri Divya, N.; Sarojini, P.; Sonika, K.: Isolation forest and local outlier factor for credit card fraud detection system. Int. J. Eng. Adv. Tech. **9**(4), 261–265 (2020)

23. Buerli, M.; Obispo, C.P.S.L.: The current state of graph databases. Dep. Computer Sci. Cal Poly San Luis Obispo mbuerli@ calpoly Edu **32**(3), 67–83 (2012)
24. Cattuto, C., Quaggiotto, M., Panisson, A., Averbuch, A.: "Time-varying social networks in a graph database: a Neo4j use case." *In: First international workshop on graph data management experiences and systems*, p. 11. ACM, (2013)
25. Allen, D., Hodler, A., Hunger, M., Knobloch M., Lyon, W., Needham, M., Voigt, H.: "Understanding trolls with efficient analytics of large graphs in neo4j." BTW (2019)
26. Kolomičenko, Vojtěch., Svoboda, Martin., Holubová Mlýnková, Irena.:"Experimental comparison of graph databases." *In: Proceedings of International Conference on Information Integration and Web-based Applications and Services*, pp. 115-124. (2013)
27. Lopez-Rojas, Edgar Alonso., Stefan, Axelsson.: Banksim: A bank payments simulator for fraud detection research. *In: Proceedings 26th European Modeling and Simulation Symposium, EMSS 2014, Bordeaux, France, Dime University of Genoa*, 2014, ISBN: 9788897999324, pp. 144–152, (2014)
28. Vicknair, C., Macias, M., Zhao, Z., Nan, X., Chen, Y., Wilkins, D.: "A comparison of a graph database and a relational database: a data provenance perspective." *In: Proceedings of the 48th Annual Southeast Regional Conference*, pp. 1–6, (2010)
29. Jouili, S., Vansteenberghe, V.: "An empirical comparison of graph databases." *In: 2013 International Conference on Social Computing*, pp. 708-715. IEEE, (2013)
30. Sengupta, Srijan; Chen, Yuguo: A block model for node popularity in networks with community structure. J. R. Stat. Soc.: Ser. B (Stat. Methodol.) **80**(2), 365–386 (2018)
31. Ren, Y. Zhu, H., ZHang, J., Dai, P., Bo, L.: "EnsemFDet: an ensemble approach to fraud detection based on bipartite graph." arXiv preprint arXiv:1912.11113, (2019)
32. Kirchner, C.; Gade, J.: Implementing Social Network Analysis for Fraud Prevention. CGI Gr, Ind (2011)
33. Freeman, L.C.: Centrality in social network conceptual clarification. Soc. Netw. **1**(3), 215–239 (1978)
34. Chen, C-J., Zaeem, RN., Barber, KS.: "Statistical analysis of identity risk of exposure and cost using the ecosystem of identity attributes." *In: 2019 European Intelligence and Security Informatics Conference (EISIC)*, pp. 32–39. IEEE, (2019)
35. Bangcharoensap, P., Kobayashi, H., Shimizu, N., Yamauchi, S., Murata, T.: "Two step graph-based semi-supervised learning for online auction fraud detection." *In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 165–179. Springer, Cham, (2015)
36. Lawrence, P.; Brin, S.; Motwani, R.; Terry, W.: Bringing order to the web. Stanford InfoLab, The PageRank citation ranking, pp. 1–17 (1999)
37. Gleich, David F.: PageRank beyond the Web. SIAM Rev. **57**(3), 321–363 (2015)
38. Van Belle, Rafaäl., Mitrović, Sandra., Weerdt, Jochen De.: "Representation learning in graphs for credit card fraud detection." *In: Workshop on Mining Data for Financial Applications*, pp. 32–46. Springer, Cham, (2019)
39. Raghavan, U.N.; Albert, R.; Kumara, S.: Near linear time algorithm to detect community structures in large-scale networks. Phys. Rev. E **76**(3), 036106 (2007)
40. Peng, L., Lin, R.: "Fraud phone calls analysis based on label propagation community detection algorithm." *In: 2018 IEEE World Congress on Services (SERVICES)*, pp. 23–24. IEEE, (2018)
41. Luan, T., Yan, Z., Zhang, S., Zheng, Y.: "Fraudster detection based on label propagation algorithm." *In: 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pp. 346–353. IEEE, (2018)
42. Wheeler, R.; Aitken, S.: Multiple algorithms for fraud detection. In: Ellis, R., Moulton, M., Coenen, F. (eds.) Applications and Innovations in Intelligent Systems VII, pp. 219–231. Springer, London (2000)
43. Li, Z., Zhang, H., Masum, M., Shahriar, H., Haddad, H.: "Cyber Fraud Prediction with Supervised Machine Learning Techniques." *In: Proceedings of the 2020 ACM Southeast Conference*, pp. 176–180. (2020)
44. Magomedov, S.; Pavelyev, S.; Ivanova, I.; Dobrotvorsky, A.; Khrestina, M.; Yusubaliev, T.: Anomaly detection with machine learning and graph databases in fraud management. Int. J. Adv. Computer Sci. Appl. **9**(11), 33–38 (2018)
45. Akinyelu, Andronicus A.; Adewumi, Aderemi O.: Classification of phishing email using random forest machine learning technique. J. Appl. Math. **2014**, 1–6 (2014)
46. Prusti, D., Rath, SK.: Fraudulent Transaction detection in credit card by applying ensemble machine learning techniques." *In: 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–6. IEEE, (2019)
47. Masoumeh, Z.; Seeja, K.R.; Afshar Alam, M.: Analysis on credit card fraud detection techniques: based on certain design criteria. Int. J. Computer Appl. **52**(3), 35–42 (2012)
48. Kiran, S.; Kumar, N.; Guru, J.; Katariya, D.; Kumar, R.; Sharma, M.: Credit card fraud detection using Naïve Bayes model based and KNN classifier. Int. J. Adv. Res., Ideas Innov. Technol. **4**(3), 44–47 (2018)
49. Mishra, Mukesh Kumar., Dash, Rajashree.: "A comparative study of chebyshev functional link artificial neural network, multi-layer perceptron and decision tree for credit card fraud detection." *In: 2014 International Conference on Information Technology*, pp. 228-233, (2014)
50. Hearst, Marti A.; Dumais, Susan T.; Osuna, Edgar; Platt, John; Scholkopf, Bernhard: Support vector machines. IEEE Intell. Syst. Appl. **13**(4), 18–28 (1998)
51. Stackelberg, BV.; Avrutin, V.; Levi, P.; Schanz, M.; Wackenhut, G.: Reconstruction of dynamical systems using constructive neural networks, pp. 1–15 (2004)
52. Webber, J.:'A programmatic introduction to neo4j." *In: Proceedings of the 3rd Annual Conference on Systems, Programming, and Applications: Software for Humanity*, pp. 217–218. (2012)
53. Miller, JJ.: "Graph database applications and concepts with Neo4j." *In: Proceedings of the Southern Association for Information Systems Conference, Atlanta, GA, USA*, 2324 (S 36). (2013)