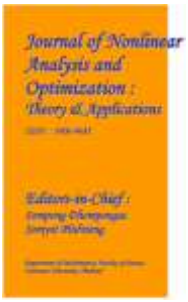
Journal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 15, Issue. 2 : 2024
ISSN : **1906-9685**



Journal of Nonlinear Analysis and Optimization : Theory & Applications

Editors-in-Chief :
Sompong Dhompongsa
Somyot Plubtieng

REPRESNTING FINE GRAINED CO-0CCURANCE BEHAVIOUR BASED FRAUID DETECTION IN ONLINE PAYMENT SERVICE

[1] VANGAVOLU PAVANI [2] MR. B. SURESH

[1] PG Scholar in the department of MCA at QIS College of Engineering & Technology (AUTONOMOUS), Vengamukapalem, Ongole- 523272, Prakasam Dt., AP., India.

[2] Professor in the department of CSE/MCA at QIS College of Engineering & Technology (AUTONOMOUS), Vengamukapalem,Ongole- 523272, Prakasam Dt., AP., India..

ABSTRACT

The vigorous development of e-commerce breeds cybercrime. Online payment fraud detection, a challenge faced by online service, plays an important role in rapidly evolving e-commerce. Behavior based methods are recognized as a promising method for online payment fraud detection. However, it is a big challenge to build high-resolution behavioral models by using low-quality behavioral data. In this work, we mainly address this problem from data enhancement for behavioral modeling. We extract fine-grained co-occurrence relationships of transactional attributes by using a knowledge graph. Furthermore, we adopt the heterogeneous network embedding to learn and improve representing comprehensive relationships. Particularly, we explore customized network embedding schemes for different types of behavioral models, such as the population-level models, individual- level models, and generalizedagent-based

models. The performance gain of our

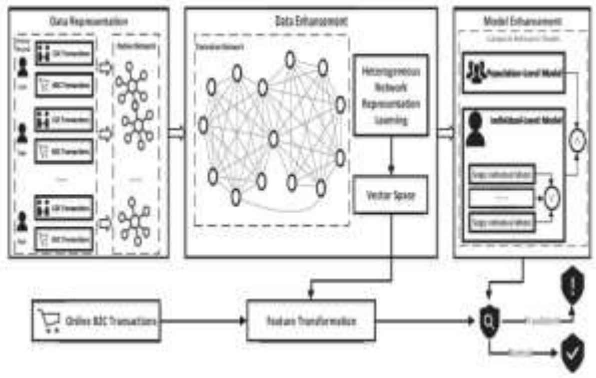
method is validated by the experiments over the real dataset from a commercial bank. It can help representative behavioral models improve significantly the performance of online banking payment fraud detection. To the best of our knowledge, this is the first work to realize data enhancement for diversified behavior models by implementing network embedding algorithms on attribute-level co-occurrence relationships.

INDEX : vigorous, development, co-occurrence, validate, attribute level, dataset.

INTRODUCTION

There are various fraudulent activities detection techniques has implemented in credit card transactions have been kept in researcher minds to methods to develop models based on artificial intelligence, data mining, fuzzy logic and machine learning. Credit card fraud detection is significantly difficult, but also popular problem to solve. In our proposed system we built the credit card fraud detection using Machine learning. With the advancement of machine learning techniques. Machine learning has been identified as a successful measure for fraud detection. A large amount of data is transferred during online transaction processes, resulting in a binary result: genuine or fraudulent. Within the sample fraudulent datasets, features are constructed. These are data points namely the age and value of the customer account, as well as the origin of the credit card.

System Architecture



METHODOLGY

ALGORITHM USED :

Decision tree classifiers

Decision tree classifiers are used successfully in many diverse areas. Their most important feature is the capability of capturing descriptive decision making knowledge from the supplied data. Decision tree can be generated from training sets. The procedure for such generation based on the set of objects (S), each belonging to one of the classes C1, C2, …, Ck is as follows:

Step 1. If all the objects in S belong to the same class, for example Ci, the decision tree for S consists of a leaf labeled with this class

Step 2. Otherwise, let T be some test with possible outcomes O1, O2,…, On. Each object in S has one outcome for T so the test partitions S into subsets S1, S2,… Sn where each object in Si has outcome Oi for T. T becomes the root of the decision tree and for each outcome Oi we build a subsidiary decision tree by invoking the same procedure recursively on the set Si.

Gradient boosting

Gradient boosting is a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees.[1][2] When a decision tree is the weak learner, the resulting algorithm is called gradient-boosted trees; it usually outperforms random forest.A gradient-boosted trees model is built in a stage-wise fashion as in other boosting methods, but it generalizes the other methods by allowing optimization of an arbitrary differentiable loss function.

K-Nearest Neighbors (KNN)

Simple, but a very powerful classification algorithm

Classifies based on a similarity measure

Non-parametric

Lazy learning

Does not “learn” until the test example is given

Whenever we have a new data to classify, we find its K-nearest neighbors from the training data

Example

Training dataset consists of k-closest examples in feature space

Feature space means, space with categorization variables (non-metric variables)

Learning based on instances, and thus also works lazily because instance close to the input vector for test or prediction may take time to occur in the training dataset

Logistic regression Classifiers

Logistic regression analysis studies the association between a categorical dependent variable and a set of independent (explanatory) variables. The name logistic regression is used when the dependent variable has only two values, such as 0 and 1 or Yes and No. The name multinomial logistic regression is usually reserved for the case when the dependent variable has three or more unique values, such as Married, Single, Divorced, or Widowed. Although the type of data used for the dependent variable is different from that of multiple regression, the practical use of the procedure is similar.

Logistic regression competes with discriminant analysis as a method for analyzing categorical-response variables. Many statisticians feel that logistic regression is more versatile and better suited for modeling most situations than is discriminant analysis. This is because logistic regression does not assume that the independent variables are normally distributed, as discriminant analysis does.

This program computes binary logistic regression and multinomial logistic regression on both numeric and categorical independent variables. It reports on the regression equation as well as the goodness of fit, odds ratios, confidence limits, likelihood, and deviance. It performs a comprehensive residual analysis including diagnostic residual reports and plots. It can perform an independent variable subset selection search, looking for the best regression model with the fewest independent variables. It provides confidence intervals on predicted values and provides ROC curves to help determine the best cutoff point for classification. It allows you to validate your results by automatically classifying rows that are not used during the analysis.

Naïve Bayes

The naive bayes approach is a supervised learning method which is based on a simplistic hypothesis: it assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature .

Yet, despite this, it appears robust and efficient. Its performance is comparable to other supervised learning techniques. Various reasons have been advanced in the literature. In this tutorial, we highlight an explanation based on the representation bias. The naive bayes classifier is a linear classifier, as well as linear discriminant analysis, logistic regression or linear SVM (support vector machine). The difference lies on the method of estimating the parameters of the classifier (the learning bias).

While the Naive Bayes classifier is widely used in the research world, it is not widespread among practitioners which want to obtain usable results. On the one hand, the researchers found especially it is very easy to program and implement it, its parameters are easy to estimate, learning is very fast even on very large databases, its accuracy is reasonably good in comparison to the other approaches. On the other hand, the final users do not obtain a model easy to interpret and deploy, they does not understand the interest of such a technique.

Thus, we introduce in a new presentation of the results of the learning process. The classifier is easier to understand, and its deployment is also made easier. In the first part of this tutorial, we present some theoretical aspects of the naive bayes classifier. Then, we implement the approach on a dataset with Tanagra. We compare the obtained results (the parameters of the model) to those obtained with other linear approaches such as the logistic regression, the linear discriminant analysis and the linear SVM. We note that the results are highly consistent. This largely explains the good performance of the method in comparison to others. In the second part, we use various tools on the same dataset (Weka 3.6.0, R 2.9.2, Knime 2.1.1, Orange 2.0b and RapidMiner 4.6.0). We try above all to understand the obtained results.

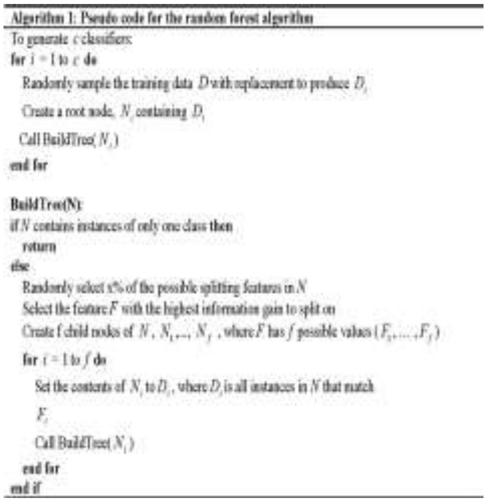
Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned. Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.

The first algorithm for random decision forests was created in 1995 by Tin Kam Ho[1] using the random subspace method, which, in Ho's formulation, is a way to implement the "stochastic discrimination" approach to classification proposed by Eugene Kleinberg.

An extension of the algorithm was developed by Leo Breiman and Adele Cutler, who registered "Random Forests" as a trademark in 2006 (as of 2019, owned by Minitab, Inc.).The extension combines Breiman's "bagging" idea and random selection of features, introduced first by Ho[1] and later independently by Amit and Geman[13] in order to construct a collection of decision trees with controlled variance.

Random forests are frequently used as "blackbox" models in businesses, as they generate reasonable predictions across a wide range of data while requiring little configuration.



Algorithm 1: Pseudo code for the random forest algorithm

```
To generate $c$ classifiers:
for $i = 1$ to $c$ do
  Randomly sample the training data $D$ with replacement to produce $D_i$
  Create a root node, $N_i$ containing $D_i$
  Call BuildTree( $N_i$ )
end for

BuildTree(N):
if $N$ contains instances of only one class then
  return
else
  Randomly select x% of the possible splitting features in $N$
  Select the feature $F$ with the highest information gain to split on
  Create f child nodes of $N$ , $N_1$ ,..., $N_f$ , where $F$ has $f$ possible values ( $F_1$ , … , $F_f$ )
  for $i = 1$ to $f$ do
    Set the contents of $N_i$ to $D_i$ , where $D_i$ is all instances in $N$ that match
    $F_i$
    Call BuildTree( $N_i$ )
  end for
end if
```

SVM

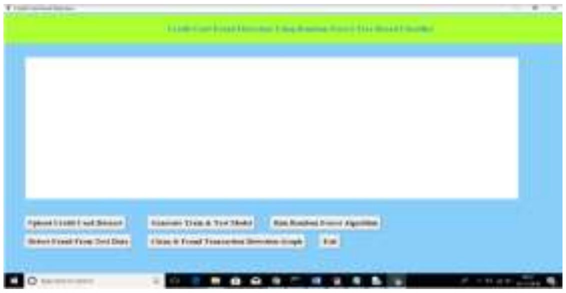
In classification tasks a discriminant machine learning technique aims at finding, based on an independent and identically distributed (iid) training dataset, a discriminant function that can correctly predict labels for newly acquired instances. Unlike generative machine learning approaches, which require computations of conditional probability distributions, a discriminant classification function takes a data point x and assigns it to one of the different classes that are a part of the classification task. Less powerful than generative approaches, which are mostly used when prediction involves outlier detection, discriminant approaches require fewer computational resources and less training data, especially for a multidimensional feature space and when only posterior probabilities are needed. From a geometric perspective, learning a classifier is equivalent to finding the equation for a multidimensional surface that best separates the different classes in the feature space.

SVM is a discriminant technique, and, because it solves the convex optimization problem analytically, it always returns the same optimal hyperplane parameter—in contrast to genetic algorithms (GAs) or perceptrons, both of which are widely used for classification in machine learning. For perceptrons, solutions are highly dependent on the initialization and termination criteria. For a specific kernel that transforms the data from the input space to the feature space, training returns uniquely defined SVM model parameters for a given training set, whereas the perceptron and GA classifier models are different each time training is initialized. The aim of GAs and perceptrons is only to minimize error during training, which will translate into several hyperplanes' meeting this requirement.

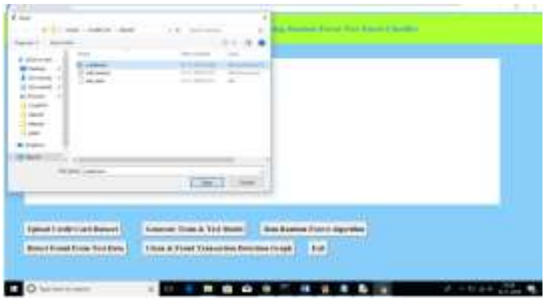
```
Algorithm 1: SVM
1. Set $Input = (x_i, y_i)$,    where $i = 1, 2, \cdots, N, x_i = R^n$ and $y_i = \{+1, -1\}$.
2. Assign $f(X) = \omega^T x_i + b = \sum_{i=1}^{N} \omega^T x_i + b = 0$
3. Minimize the QP problem as, $min\ \varphi(\omega, \xi) = \frac{1}{2}\|\omega\|^2 + C.(\sum_{i=1}^{N} \xi_i)$.
4. Calculate the dual Lagrangian multipliers as $min\ L_p = \frac{1}{2}\|\omega\|^2 - \sum_{i=1}^{N} x_i y_i(\omega x_i + b) + \sum_{i=1}^{N} x_i$ .
5. Calculate the dual quadratic optimization (QP) problem as $max\ L_D = \sum_{i=1}^{N} x_i - \frac{1}{2}\sum_{i,j=1}^{N} x_i x_j\ y_i\ y_j (x_i, x_j)$.
6. Solve dual optimization problem as $\sum_{i=1}^{N} y_i x_i = 0$ .
7. Output the classifier as $f(X) = sgn(\sum_{i=1}^{N} x_i y_i (x \cdot x_i) + /$.
```

RESULTS ANALYSIS

To run project double click on 'run.bat' file to get below screen

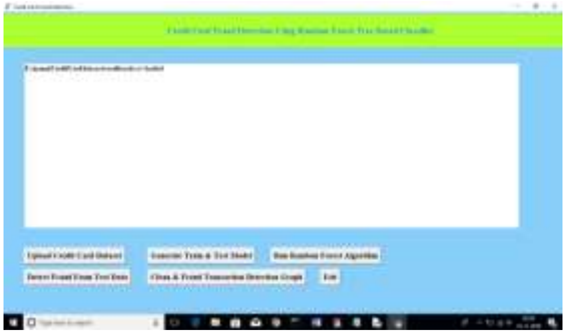


In above screen click on 'Upload Credit Card Dataset' button to upload dataset



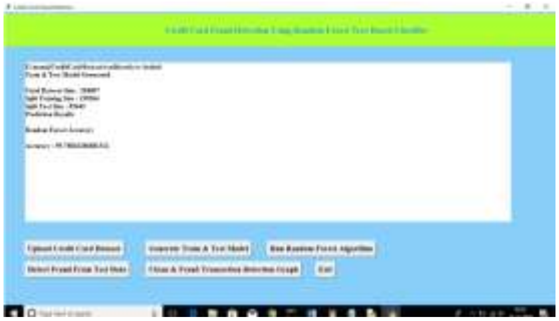
REPRESENTING FINE GRAINED CO-OCUURANCE BEHAVIOUR BACSD FROUD DECETION IN ONLINE PAYMENT SERVICE

After uploading dataset will get below screen



Now click on 'Generate Train & Test Model' to generate training model for Random Forest Classifier

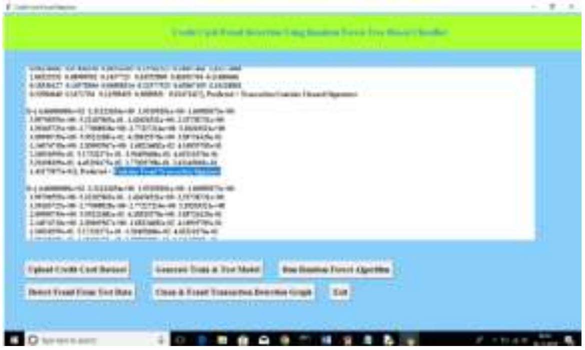
In above screen after generating model we can see total records available in dataset and then application using how many records for training and how many for testing. Now click on "Run Random Forest Algorithm' button to generate Random Forest model on train and test data


Upload Credit Card Dataset
Generate Train & Test Model
Run Random Forest Algorithm
Detect Fraud From Test Data
Clean & Fraud Transaction Detection Graph
Exit

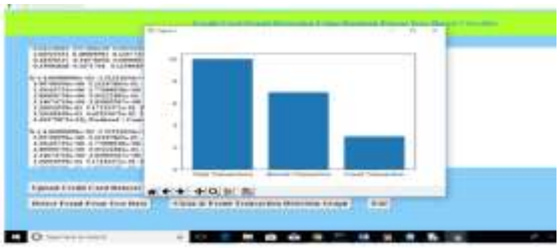
In above screen we can see Random Forest generate 99.78% percent accuracy while building model on train and test data. Now click on 'Detect Fraud From Test Data' button to upload test data and to predict whether test data contains normal or fraud transaction

Upload Credit Card Dataset
Generate Train & Test Model
Run Random Forest Algorithm
Detect Fraud From Test Data
Clean & Fraud Transaction Detection Graph
Exit

In above screen I am uploading test dataset and after uploading test data will get below prediction details


Upload Credit Card Dataset
Generate Train & Test Model
Run Random Forest Algorithm
Detect Fraud From Test Data
Clean & Fraud Transaction Detection Graph
Exit

Now click on 'Clean & Fraud Transaction Detection Graph' button to see total test transaction with clean and fraud signature in graphical format. See below screen



In above graph we can see total test data and number of normal and fraud transaction detected. In above graph x-axis represents type and y-axis represents count of clean and fraud transaction

CONCLUSION

The Random forest algorithm will perform better with a larger number of training data, but speed during testing and application will suffer. Application of more pre-processing techniques would also help. The SVM algorithm still suffers from the imbalanced dataset problem and requires more preprocessing to give better results at the results shown by SVM is great but it could have been better if more preprocessing have been done on the data

Future enhancement

As online payment services continue to evolve and adapt to changing consumer behaviors and technological advancements, the future of fraud detection lies in further refinement and innovation of fine-grained cooccurrence behavior-based approaches. One promising direction is the integration of advanced machine learning techniques, such as deep learning and reinforcement learning, to enhance the accuracy and efficiency of fraud detection models. These techniques offer the potential to capture complex patterns and dynamics of fraudulent behavior in online transactions, leading to more effective detection and prevention of fraudulent activities. Additionally, the incorporation of real-time data streams from diverse sources, such as user interactions, device attributes, and transaction metadata, can enrich the feature space and provide deeper insights into fraudulent behavior patterns.

REFERENCES

[1] SERVIC Sudhamathy G: Credit Risk Analysis and Prediction Modelling of Bank Loans Using R, vol. 8, no5, pp. 1954-1966.

[2] LI Changjian, HU Peng: Credit Risk Assessment for ural Credit Cooperatives based on Improved Neural Network, International Conference on Smart Grid and Electrical Automation vol. 60, no. - 3, pp 227-230, 2017.

[3] Wei Sun, Chen-Guang Yang, Jian-Xun Qi: Credit Risk Assessment in Commercial Banks Based On Support Vector Machines, vol.6, pp 2430-2433, 2006.

[4] Amlan Kundu, Suvasini Panigrahi, Shamik Sural, Senior Member, IEEE, “BLAST-SSAHA Hybridization for Credit Card Fraud Detection”, vol. 6, no. 4 pp. 309-315, 2009.

[5] Y. Sahin and E. Duman, “Detecting Credit Card Fraud by Decision Trees and Support Vector Machines, Proceedings of International Multi Conference of Engineers and Computer Scientists, vol. I, 2011.

[6] Sitaram patel, Sunita Gond , “Supervised Machine (SVM) Learning for Credit Card Fraud Detection, International of engineering trends and technology, vol. 8, no. -3, pp. 137- 140, 2014.

[7] Snehal Patil, Harshada Somavanshi, Jyoti Gaikwad, Amruta Deshmane, Rinku Badgujar," Credit Card Fraud Detection Using Decision Tree Induction Algorithm, International Journal of Computer Science and Mobile Computing, Vol.4 Issue.4, April- 2015, pg. 92-95

[8] Dahee Choi and Kyungho Lee, “Machine Learning based Approach to Financial Fraud Detection Process in Mobile Payment System", vol. 5, no. - 4, December 2017, pp. 12-24

AUTHOR PROFILE:

Ms. VANGAVOLU PAVANI currently pursuing Master of Computer Applications at QIS College of engineering and Technology (Autonomous), Ongole, Andhra Pradesh. He Completed B.Sc. in Computer science from acharya nagarjuna university, Andhra Pradesh. His area of interest is Machine Learning, Cloud Computing and DevOps

.Mr.B.Suresh, currently working as an Associate Professor in the Department of Master of Computer Applications, QIS College of Engineering and Technology, Ongole, Andhra Pradesh. His area of interest is Machine Learning, Cloud Computing and Programming Languages