Real-time credit card fraud detection using Salient Feature Extraction Technique with Adaptive Synthetic Oversampling Model

***Abstract*—Credit card fraudulence is a federal offense that takes place frequently in recent times. The phenomenon where an imposter or a scammer tries to make a purchase or transfer money from one account to another using a credit card that does not belong to him/her, is coined as Credit Card Fraudulence. In modern world, credit card fraud or any type of payment card fraud is a very common but serious crime that occurs both offline and online. But with the help of machine learning algorithms and Salient Feature Extraction Technique (SFET) we can easily detect such offense and help in further investigations. From time to time many data scientists, data analysts, machine learning engineers and other researchers have designed many algorithms to detect credit card frauds. By extracting the most relevant and important features of a transaction, it is quite possible to detect credit card fraud very quickly & efficiently. In this paper, we have shown such an improved way by using Adaptive Synthetic oversampling (ADASYN) model with five notable supervised machine learning models namely Random Forest, Support Vector Machine, Naive Bayes, Logistics Regression and K-Nearest Neighbour. Out of these five machine learning models, K-Nearest Neighbour has shown the best precision, recall, specificity & accuracy. The performance accuracy of Random Forest, Logistic Regression, K-Nearest Neighbour, Naive Bayes & Support Vector Machines are 96.04%, 81.31%, 96.22%, 79.22% & 50.06% respectively.**

***Index Terms*—Precision, Recall, Specificity, fraudulence, target-variable, decision-trees, ensemble, outlier**

I. Introduction

The act of using third party credit card information for product purchase or bank balance transaction is commonly known as credit card transaction fraudulence. Credit card fraud is an increasing federal crime that occurs in both online and offline bank transactions. In the world of digital and online banking, scammers and fraudsters are always ready to grab the chance of committing this offense. And it has been quite difficult to identify a fraud transaction due to lack of enough digital evidence on which the commonly used machine learning algorithms can rely [1]. In this paper, we have used a simple but sophisticated technique of feature extraction in the name of Salient Feature Extraction Technique (SFET) using which a supervised machine learning algorithm will extract or convert a given transaction feature to a relevant and machine readable format and train itself.

Supervised Machine Learning algorithms like Random Forest (RF), Support Vector Machine (SVM), Logistics Regression (LR), Naive Bayes (NB) and K-Nearest Neighbour (KNN) work best on supervised datasets. In this case, the transaction records of credit card plays the role of a supervised dataset as these records are always organized and labelled.

So, in this paper, we have simulated a supervised transaction dataset in five supervised machine learning algorithms that firstly classifies the given data, trains itself with the data and lastly validates (regression) its learning. For the model training purpose we allocated 70% of the data and the remaining 30% was used for validation. Our key contributions in this work can be summarized as follows:

- To successfully apply ADASYN (Adaptive Synthetic Sampling Method) and Stratified K-Fold Cross Validation to deal with the class (fraud & non-fraud) imbalance of our dataset.
- To use an efficient feature extraction process with the name of Salient Feature Extraction Technique (SFET) which is a very simple but efficient way of converting and extracting data features into machine readable format.
- To successfully administer the model simulations and gained precision, recall, specificity and accuracy scores.

This paper consists of eight sections. Section II represents the related works and conducted experiments relating to our simulations by other researchers. In Section III a brief description of our simulated machine learning models is given. Section IV deals with the dataset and simulator that we used. The concept and brief discussion of our simulation procedure, dataset shaping, noise reduction and Salient Feature Extraction Technique (SFET) is offered in Section V. Section VI narrates the relevant experimental results while the result analysis with proper comparison and discussion is shown in Section VII. Lastly, we concluded the paper in Section VIII.

II. Related Works

Notable researchers and data scientists have conducted several simulations on real time credit card fraud detection using supervised machine learning models like Isolation Forest, Random Forest, Decision Trees and much more. N. K. Trived, S. Simaiya [3] represented the accuracy of Random Forest, SVM, KNN, Logistics Regression & Naive Bayes as 94.99%, 93.96%, 94.99%, 90.45% & 91.89% respectively. Similarly, S. Shirgave, R. More [5] discussed their accuracy for the same models to be 96.2%, 93.8%, 94.2%, 94.7% & 93.7% respectively in their paper. I. Sadali, N. Sael & F. Benabbou [8] simulated Support Vector Machine, Random Forest & K-Nearest Neighbour for their comparative research study and their obtained model accuracy were 99.7%, 82.5% & 97.1% respectively. Some researchers simulated & compared their model accuracy on the basis of using and not using ’transaction time' as a feature [4]. As such their differentiated accuracy for the five aforementioned algorithms with 'transaction time' feature were 93.9% (Random Forest), 93.6% (SVM), 92.6% (KNN), 93.9% (Logistics Regression) & 90.9% (Naive Bayes).

III. Supervised Machine Learning Algorithms

Supervised machine learning is the learning process of a function that maps an input to an output based on the given input-output pairs. Supervised learning requires both supervised algorithms and supervised datasets. The main advantage of using supervised learning is that it allows data prediction judging from previous experience (simulated dataset). In a supervised dataset, the classes, features and the target variable are sorted and labelled, as such classification and prediction becomes easy for the used learning model. In our simulation, we have used five notable and commonly-used supervised machine learning algorithm.

A. Random Forest

Random Forest algorithm is mostly a decision tree-based algorithm where it generates multiple decision trees to improve the output by combining it with the generalization capacity of the model. The process of combining trees is called an ensemble strategy which is a combination of individual trees to create a solid learner [9]. It can be utilized to deal with regression and classification issues. In regression issues, the dependent variable is continuous while in classification issues, the subordinate variable is categorical.

B. Support Vector Machine

The Support Vector Machine was first introduced by Vapnik (1995). It is a supervised training algorithm which is capable of deciphering subtle patterns in complex datasets. This learning model can be used for both classification and regression purposes. SVM modeling includes two stages, to train data set and plot a model utilize and later use this model to predict information of a test data set [6].

C. Naive Bayes

Bayesian network classifiers are exceptionally imperative within the region of machine learning and its under the category of directed classification models. The Naïve Bayes machine learning classifier tends to discover a course which is known as 'result course' based on probabilities of occurrences [2]. As such, the nature of this learning is very proficient, swift and straight in exactness for real-world scenarios.

D. Logistics Regression

Logistic Regression is an administered learning algorithm that determines the likelihood of the dependent variable from the independent variable of a dataset. It produces logistic curves which plot the values somewhere in the range of 0 and 1. It is a regression model where the dependent variable is categorical and cannot function with outliers in the dataset [7]. The model of logistics regression is:

$$\log(\frac{p}{1-p}) = \beta_o + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k \qquad (1)$$

Here, p denotes the probability of the independent variable to be 1, β_0 is a constant and $\beta_1 ... \beta_k$ are the co-efficients of independent variables. $X_1 ... X_k$ are independent variables.

E. K-Nearest Neighbour

The concept of the nearest neighbor analysis has been utilized in various anomaly or outlier detection techniques. It is a supervised learning algorithm where the result of a new occurrence query is characterized depending on the majority of the K-nearest neighbor category [10]. The performance of the KNN algorithm is influenced by the following three principal factors:

- The distance metric is used to find the closest neighbors.
- The distance rule is used to derive a classification from the K-nearest neighbor.
- The number of neighbors determined is later used to group the new sample.

IV. Dataset & Simulator

A. Dataset

The dataset [11] we used is a simulated credit card transaction dataset that contains both legitimate and fraud transactions from the duration of 1st January 2019 to 31st December 2020 (2 years). It covers the transaction records of 1000 customers doing transactions with a pool of 800 merchants.The dataset contains data of exactly 16,04,294 transactions of which 8,151 are frauds and 15,96,143 are legitimate. Fig.(1) illustrates the graphical analysis of the dataset features.

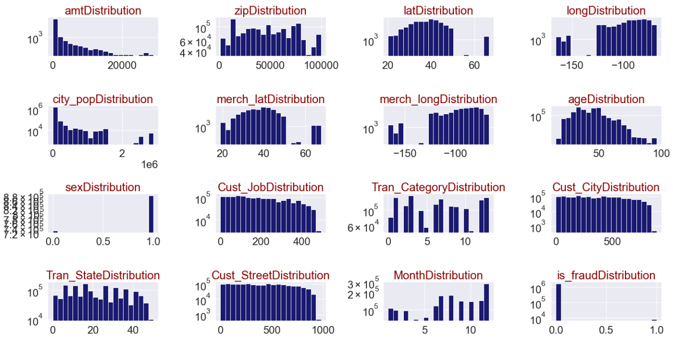

Fig. 1. Dataset Description

B. Simulation Environment

We conducted our simulation in an IDE named Jupyter Notebook. Our used programming language was Python version 3.9.1. To conduct the simulation, we needed some very important python libraries like Numpy, Pandas, Matplotlib, Seaborn, SciKit, Datetime and more. We imported the necessary libraries in the IDE before starting the simulations.

V. Methodology

As mentioned earlier, we have conducted our simulations using five supervised machine learning models named Random Forest, Logistics Regression, K-Nearest Neighbour, Support Vector Machine & Naive Bayes. In order to get optimum accuracy, we have to shape the dataset properly, reduce unnecessary and missing data, remove outliers or anomalies, extract important data from the given data and much more.

A. Working Procedure

Fig.(2) represents the workflow we followed to determine the best machine learning model for efficient credit card fraudulent detection.

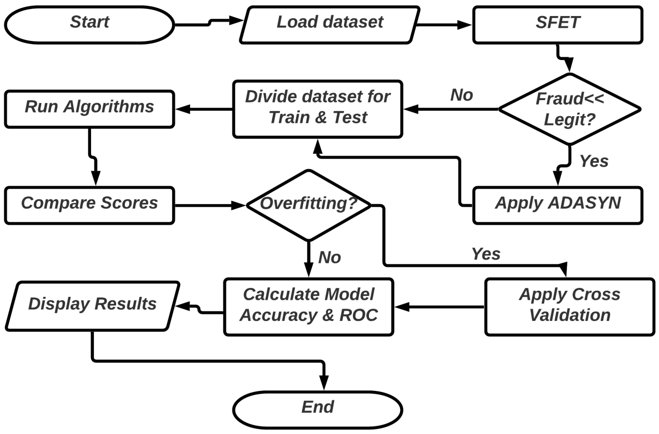

Fig. 2. Implemented workflow

B. Dataset Processing

As it is quite clear from the aforementioned dataset distribution that our used dataset is highly imbalanced, we had to use oversampling to balance out the classes. In our dataset, there are two classes namely Fraud & Non-Fraud. Initially the dataset contained 23 columns out of which 22 denoted relevant & salient features of the transactions while the last column was the class column that represented whether the transactions were fraud or legitimate. This column is called label or target variable.

First, we extracted some very important but missing features from the existent ones such as, 'age' from 'DOB', 'month' from 'trans_date_trans_time' as well as mapped some existing features to numbers for the machine readability. We dropped the irrelevant columns (features) such as 'transaction id', 'cc num', 'name of the cardholder' and such. To do such, we graphed the correlation of the features with the target variable first. Fig.(3) demonstrates the graphical correlation of the features with the label. Secondly, we divided the dataset into X & Y representing 'Features' & 'Label' respectively. The whole dataset was then divided into training (70%) and testing (30%) sets maintaining an equal distribution of the classes between the datasets.

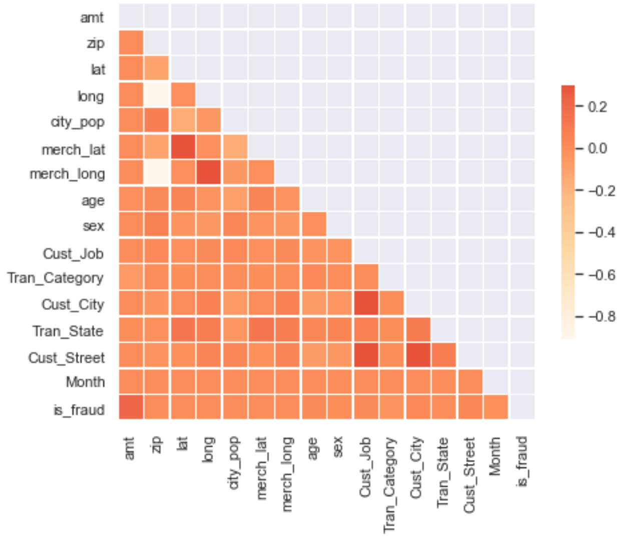

Fig. 3. Correlation between features & label

In order to overcome the class imbalance of our dataset, we applied Adaptive Synthetic Sampling Model (ADASYN) that over-sampled the minority class (Fraud) and made both classes equal. As such, the machine learning models underwent equal number of fraud and legitimate transactions in the training process. Furthermore, we faced the problem of over-fitting training & validation scores by applying stratified k-fold cross validation.

C. Salient Feature Extraction Technique (SFET)

Extracting the most salient & relevant features of a transaction is a very important task in credit card fraud detection. The dataset used to train the models may contain many unnecessary features like 'Customer Name', 'Card Number', 'Transaction Number', 'Date of Birth' and more that does not have any adherence with the fraudulence. Since the owner of the card or the card itself has nothing to do with the fraud, any features indicating the owner's personal credentials should be ignored. Rather the features relating to the transactions, customers and merchants should be used. Table I contains detailed information of the features we converted and extracted for model training and validation purposes.

D. Model Parameters

In our model simulation, we have used various model parameters to manipulate the simulations. Random Forest, Logistic Regression, Support Vector Machine, Naive Bayes & K Nearest Neighbour use different parameters for proper classification and regression purpose. Table (II) shows the respective parameters of the models we used for our simulations.

TABLE I
SALIENT FEATURES WITH DESCRIPTIONS

Sl.	Features	Descriptions	Status
1.	amt	Amount of Transaction	Given
2.	zip	Zip Code of Customer	Given
3.	lat	Latitude of Customer	Given
4.	long	Longitude of Customer	Given
5.	merch_lat	Latitude of Merchant	Given
6.	merch_long	Longitude of Merchant	Given
7.	city_pop	Population of the Transaction City	Given
8.	Cust_Job	Job of the Customer	Converted
9.	Tran_Category	Type of the purchased product	Converted
10.	Cust_City	City of Customer	Converted
11.	Tran_State	State where transaction occurred	Converted
12.	Cust_Street	Street of the Customer	Converted
13.	Month	Month of Transaction	Extracted
14.	age	Age of Customer	Extracted
15.	sex	Sex of Customer	Converted

TABLE II
PARAMETERS OF MODELS

Model	Parameters
Random Forest	criterion = ’gini’, n_estimators=200
Logistic Regression	max_iter=200
Support vector Machine	kernel= ’rbf’, max_iter=200, gamma=’auto’
Naive Bayes	Null
KNN	n$_neighbors = 5, n\_jobs = 16$

VI. EXPERIMENTAL RESULTS

The results we achieved from our simulations can be divided into two categories i.e. before oversampling & after oversampling. We achieved individual data training, testing results along with respective confusion matrices, precision scores, recall scores, F1-scores & accuracy.

A. *Dataset training & validation results*

Fig.(4) shows the training & testing results of the models before applying Adaptive Synthetic Oversampling.

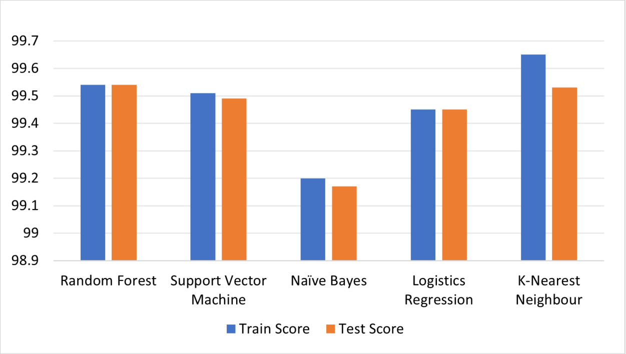

Fig. 4. Train & Test Scores of the models (before ADASYN)

B. *Precision, Recall, Specificity & Accuracy of the models*

It is quite difficult to understand the efficiency and performance of a model from its accuracy alone. So, we have used multiple performance metrics namely precision, recall, specificity & accuracy to evaluate the models. Fig.(5) compares the precision, recall, specificity & accuracy of the models before applying ADASYN while Fig.(6) shows the silimar metrics comparison of the models after ADASYN. Table(III) & Fig.(7) depicts the accuracy comparison of the models before and after applying ADASYN.

TABLE III
MODEL ACCURACY

Model	Before ADASYN	After ADASYN
Random Forest	99.54	96.04
Logistic Regression	99.45	81.31
Support vector Machine	99.49	50.06
Naive Bayes	99.17	79.22
KNN	99.53	96.22

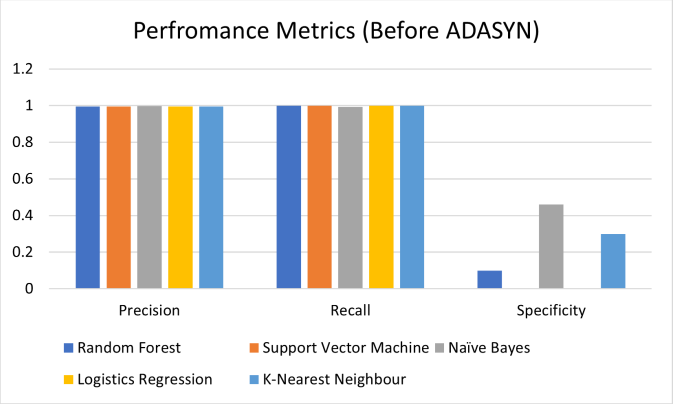

Fig. 5. Model Precision, Recall, Specificity (before ADASYN)

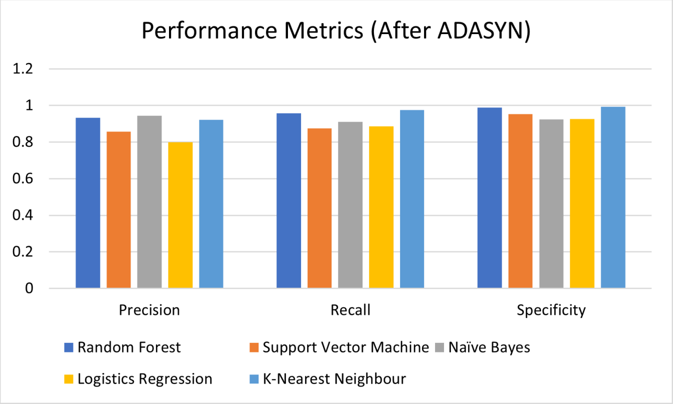

Fig. 6. Model Precision, Recall, Specificity (after ADASYN)

VII. RESULT ANALYSIS

A. *Comparison*

From our model simulations, we have found that K Nearest Neighbour has given the best performance among all the models. The accuracy, precision, recall and specificity of KNN after applying ADASYN are 96.22%, 92.1%, 97.6%, 99.4% respectively. Taking all the metrics in consideration, KNN

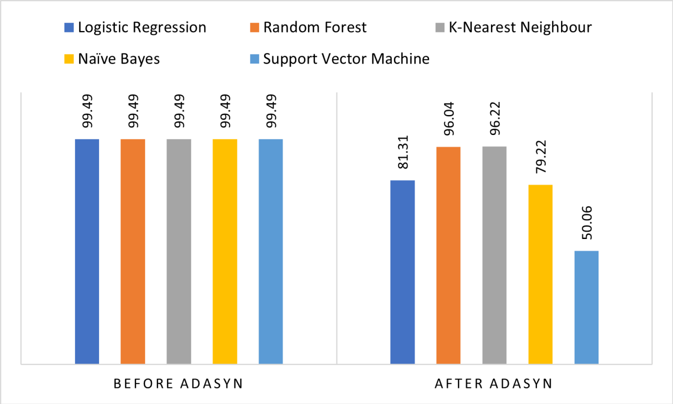

Fig. 7. Models Accuracy before & after ADASYN

gives the best performance than Random Forest, Logistic Regression, Support Vector Machine & Naive Bayes.

Furthermore, we have compared our obtained results with some of the relevant and notable research on credit card fraud detection. Table (IV) & Fig. (8) shows the comparison.

TABLE IV
PARAMETERS OF MODELS

Paper Reference	Best Algorithm	Accuracy
[3]	KNN	94.99
[4]	Random Forest	93.9
[5]	Random Forest	96.2
[7]	Logistic Regression	95.15
[8]	SVM	99.7
Our Model	KNN	96.22

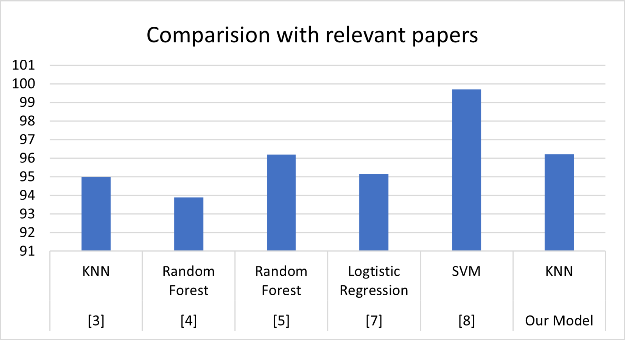

Fig. 8. Comparison with relevant papers

B. Discussion

From Table(IV), it is quite evident that our best model KNN has shown good performance and it efficient in detecting fraud transactions. With Stratified K-Fold cross validation and Adaptive Synthetic Oversampling, the dataset imbalance can be easily nullified and the models can be efficiently trained. K Nearest Neighbour has good performance and control over supervised datasets and is one of the best supervised machine learning algorithm in the market.

VIII. Conclusion

With proper data pruning, noise reduction, feature extraction, need-based oversampling, cross-validation and model training real time credit card fraudulence detection is a possible thing in today's world. If we can handle the imbalance dataset with any oversampling or undersampling technique, we can easily get efficient results in differentiating fraud credit card transactions (both purchase & money-transfer) from legitimate ones. Although our model has shown significant performance in detecting frauds, the performance of algorithms differ with the used method of dataset balance. As such, to get good performance and output from K Nearest Neighbour, stratified K-fold & Adaptive Synthetic Oversampling must be ensured.

References

[1] Elsevier B. V.,"Credit Card Fraud Detection Using Machine Learning Algorithms", International Conference on Recent Trends in Advanced Computing 2019, ICRTAC 2019, Procedia Computer Science 165 (2019), pp. 631-641

[2] N. Khare, S. Y. Sait, "Credit Card Fraud Detection Using Machine Learning Models and Collating Machine Learning Models", International Journal of Pure and Applied Mathematics, Vol 118, No. 20, 2018, pp. 825-838

[3] N. K. Trivedi, S. Simaiya, U. K. Lilhore, S. K. Sharma, "An Efficient Credit Card Detection Model Based On Machine Learning Models", International Journal of Advanced Science and Technology, Vol 29, 2020, pp. 3414-3424

[4] S. Rajora, D. L. Li, C. Jha, N. Bharill, O. P. Patel. S. Joshi, D. Puthal, M. Prasad, "A Comparative Study of Machine Learning Techniques for Credit Card Fraud Detection Based on Time Variance", IEEE Symposium Seris on Computational Intelligence SSCI 2018, pp. 1958-1963

[5] S. K. Shirgave, C. J. Awati, R. More, S. S. Patil, "A Review On Credit Card Fraud Detection Using Machine Learning, "International Journal of Scientific & Technology Research, vol 8, Issue 10, October 2019, ISSN 2277-8616

[6] D. Varmedja, M. Karanovic, S. Sladojevic, M. Arenovic, A. Anderla, "Credit Card Fraud Detection Using Machine Learning Methods", 18th International Symposium INFOTEH-JAHORINA, 20-22 March 2019

[7] Y. Jain, N. Tiwari, S. Dubey, S. Jain, "A Comparative Analysis of Various Credit Card Fraud Detection Techniques", International Journal of Recent Technology and Engineering (IJRTE), ISSN: 2277-3878, Vol 7, ISSUE-5S2, January-2019

[8] I. Sadagali, N. Sael, F. Benabbou, "Fraud Detection in Credit Card Transaction using Machine Learning Techniques", 9th International Conference on Cloud Computing, Data Science and Engineering, 2019

[9] S. Mittal, S. Tyagi, "Performance Evaluation of Machine Learning Algorithms for Credit Card Fraud Detection", 9th International Conference on Cloud Computing, Data Science and Engineering, 2017

[10] S. Kiran, J. Guru, R. Kumar, N. Kumar, D. Katariya, M. Sharma, "Credit Card Fraud Detection using Naive Model Based and KNN Classifier", International Journal of Advance Research, Ideas and innovations in Technology, ISSN: 2454-132X, Vol 4, Issue 3

[11] K. Shenoy, "Credit Card Transactions Fraud Detection," retrieved from https://www.kaggle.com/kartik2112/fraud-detection