Implementation of Credit Card Fraud Detection System with Concept Drifts Adaptation

Anita Jog and Anjali A. Chandavale

Abstract There is a large number of credit card payments take place that is targeted by fraudulent activities. Companies which are responsible for the processing of electronic transactions need to efficiently detect the fraudulent activity to maintain customers' trust and the continuity of their own business. In this paper, the developed algorithm detects credit card fraud. Prediction of any algorithm is based on certain attribute like customer's buying behavior, a network of merchants that customer usually deals with, the location of the transaction, amount of transaction, etc. But these attribute changes over time. So, the algorithmic model needs to be updated periodically to reduce this kind of errors. Proposed System provides two solutions for handling concept drift. One is an Active solution and another one is Passive. Active solution refers to triggering mechanisms by explicitly detecting a change in statistics. Passive solution suggests updating the model continuously in order to consider newly added records. The proposed and developed system filters 80% fraudulent transactions and acts as a support system for the society at a large.

Keywords Credit card · Concept drift adaptation · Credit card fraud detection Network-based extension

1 Introduction

Credit cards and debit cards are used extensively in today's world. People prefer executing banking transaction online rather than going to the branch. This is due to easy availability of modern resources like laptops, phone, and tablets. Most of the banks switched to centralize processing to support this trend. Another reason for credit card gaining popularity is online shopping trend. Lots of deals, promotions

A. Jog (✉) · A. A. Chandavale
Department of Information Technology, MIT College of Engineering, Pune, India
e-mail: Anita.visal@gmail.com

A. A. Chandavale
e-mail: Anjali.chandavale@mitcoe.edu.in

S. Bhalla et al. (eds.), *Intelligent Computing and Information and Communication*,
Advances in Intelligent Systems and Computing 673,
https://doi.org/10.1007/978-981-10-7245-1_46

attract customers to buy online. Online shopping allows customers to explore more items in short time. Customers can also compare the prices of the same item from different vendors. All these online activities are more prone to fraudulent cases due to lack of proper training or awareness before using digital transaction. During online transactions, personal data like account number, password, birth date, credit card details, etc. is exposed over the network resulting in financial fraud. Financial fraud causes loss of huge amount worldwide each year. There is a huge amount of loss occurs due to online fraud. As per cybercrime report [1], total card sales volume is $28.844 trillion while the losses occurred from fraud on cards is $16.31 billion. Comparing to past year fraud increased by 19%, while the overall volume of card sales only grew by only 15%. Companies that are responsible for processing electronic transactions need to efficiently detect the fraudulent activity to maintain customers' trust and the continuity of their business. Data mining offers a number of techniques to analyze patterns in data, differentiating genuine transaction from suspicious transactions. The most important aspect of fraud detection is to correctly identify the atypical character of fraud, since the fraudulent transactions are very few as compared to the legitimate transactions. A critical step in an efficient fraud detection process is creating a set of significant characteristics that capture irregular behavior. The detection of fraud is a complex computational task and there is no ideal system that accurately predicts fraudulent transaction. These systems only predict the probability of the transaction to be fraudulent. The paper is divided into following sections. Section 2 presents the related work. Section 3 describes the proposed work. In Sect. 3.3.3 expected result is outlined and Sect. 4 describes the conclusion.

2 Related Work

Various data mining techniques exist for fraud detection [2, 3, 4, 5]. Support Vector machine, genetic programming, Artificial Intelligence, Neural networks, Decision tree are few of them [6, 7, 8, 9]. Using decision tree, the system is described by Yusuf Sahin and Serol Bulkan. In this paper, cost-sensitive decision tree approach is suggested which minimizes the sum of misclassification costs while selecting the splitting attribute at each nonterminal node [10]. Author has compared the performance of this approach with the well-known traditional classification models on a real-world credit card data set. Andrea Dal Pozzolo, Giacomo Boracchi designed two fraud detection systems using ensemble and sliding window approach [11]. Customer feedbacks and alerts provided by the system are used as supervised samples. One more novel approach is described by Véronique Van Vlasselaer and Cristián Bravo [12]. This approach is a combination of the characteristics of incoming transactions along with the customer spending history based on the fundamentals of RFM (Recency–Frequency–Monetary); and characteristics by exploring the network of credit card holders and merchants. A time-dependent suspiciousness score is calculated for each network object. As seen from the

literature, it is essential to have a more efficient system capable of detecting the frauds at the very first instance. The following section describes the proposed and developed system based on the concept drift adaption which values the customer feedback. It performs a proactive heuristic search across all the customers.

3 Proposed Work

The proposed and developed system works on the principle of the active solution to detect the fraud. It follows the layered architecture namely data layer, utility layer, manager layer, and controller layer. The data layer is responsible for storing the historical data, transactional data, the model which contains the set of attributes to validate for each customer. Utility layer acts as a support layer. Manager layer is responsible for executing each job separately while controller encapsulates all the flows for every user action. The concept drift adaptation technique is implemented by invoking the scheduler periodically to update the model. Below section covers the overview of how credit card processing happens followed by the technical details of the proposed system.

3.1 Overview of Credit Card Processing

1. *Authorization*: It is the first step in which a merchant swipes the card. Then, the login details are then passed to an acquirer. Acquirer authorizes the transaction. The acquirer is nothing but a bank. It is responsible for processing and settling credit card transactions by consulting a card issuer. The acquirer then sends the request to the card-issuing bank. Card-issuing bank either authorizes or denies the request. Depending upon the decision the merchant is allowed to proceed with the sale.
2. *Batching*: Merchant reviews all the day's sales to make sure all of them were authorized and endorsed by the cardholder. It then submits all the transactions in a batch, to the acquirer to process payment.
3. *Clearing*: Once acquirer receives the batch, it sends it through the card network. Card network is responsible for routing each transaction to the appropriate issuing bank as shown in Fig. 2. The issuing bank then deducts its interchange fees and sends the remaining amount to acquirer through the network. The interchange fee is the amount paid by merchants to a credit card issuer and a card network for accepting credit cards. It usually ranges between 1 and 3%. Card network is Visa, MasterCard, or other networks that act as a mediator between an acquirer and an issuer. They are responsible for authorizing credit card transactions.

4. *Funding*: This is the last step in processing a credit card. Acquirer deducts its discount charges and sends the remaining amount to the merchant after receiving payment from the issuer. Thus, the merchant got paid for the transaction. Thus, the cardholder is charged. Merchants pay a discount fee to acquirers to incur the processing cost of the credit cards payment.

3.2 Proposed System

Figure 1 shows the block diagram of the proposed system. Using the historical transactional data, the model builder will the model. This model is nothing but the set of differentiating attributes for identifying the fraudulent transaction. This model is used by online fraud detector to decide if each incoming transaction contains any suspicious characteristic. Based on the decision given by online fraud detector, transaction processor either aborts the transaction or proceeds for further processing. Separate scheduler which runs daily triggers the offline calibrator. The offline calibrator is responsible for correcting the model as per customer feedback. This model is updated weekly to consider newly added transactions. The application has following modules starting from user login, viewing account statement, downloading in PDF or excel format to actual transaction execution which covers different validations. Some validations are performed during actual transaction while some heuristic checks are applied across all customers, where the pattern is recognized and suspicious transactions are marked in the database table.

As shown in Fig. 2, the proposed system is broadly divided into four layers

1. *Controller Layer*: It controls the flow of actions. It calls appropriate manager to execute each action. It calls alert utility depending upon the output received from manager layer. For every user action, there is one controller class. It decides different steps that need to be performed to accomplish given task.
2. *Manager Layer*: It executes below described action using database tables. It sends notifications, alerts to the controller.
    (i) *Authentication Manager*: First calls authentication manager to verify the userId, password. If the authentication manager successfully returns as a valid user, then the transaction controller passes the request to the preliminary validator.
    (ii) Fraud detection manager: After passing all the validation checks, the controller calls fraud detection manager to detect any unusual behavior of transaction. It performs few fraud detection checks which include amount, location, valid merchant, last transaction time, etc.
    (iii) Transaction Manager: Finally, the controller calls Transaction manager to execute the transaction dealing with debit/credit, transfer, etc.
3. *Utility Layer*: It takes care of reporting. Notifications, Alerts triggered by manager layer are converted into appropriate Actions.

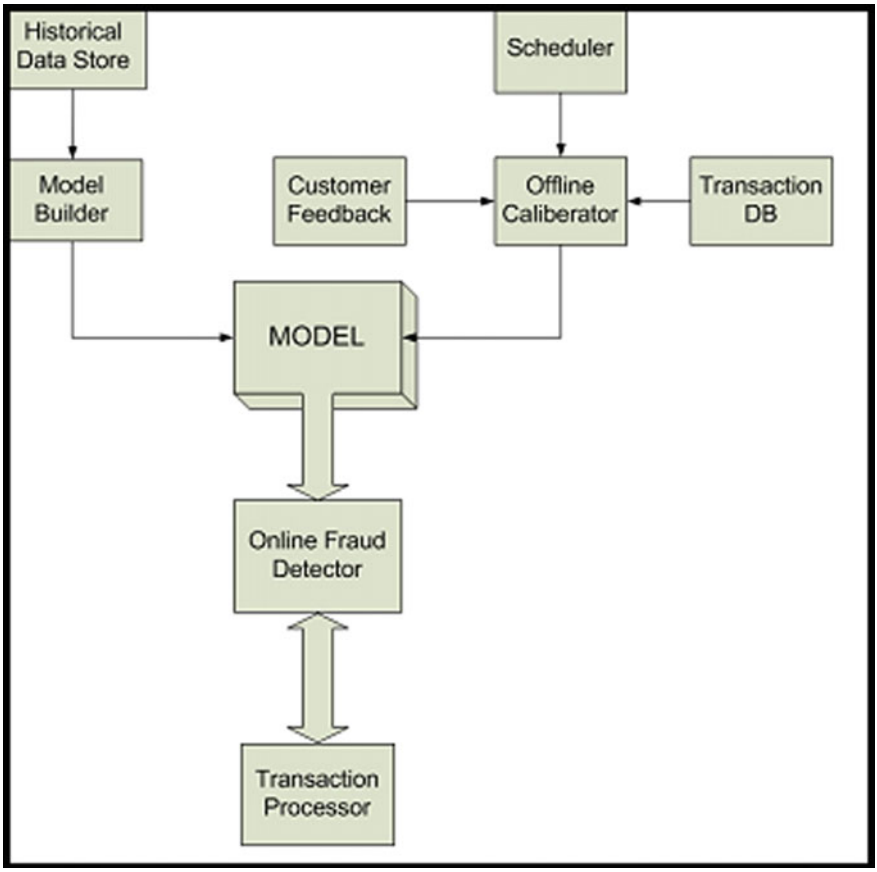

Fig. 1 Block diagram

4. *Database Layer*: It consists of transactional data, historical data, rules model data. Transactions flagged by the algorithm kept in the separate table. Rules Model table is populated based on historical data. And, it gets updated periodically using self-learning algorithm. The following figure shows a quick overview of the database layer.

3.3 Implementation

The system is implemented in JAVA, JSP. The model is determined by WEKA output. SQL server database is used for backend data storage. For download

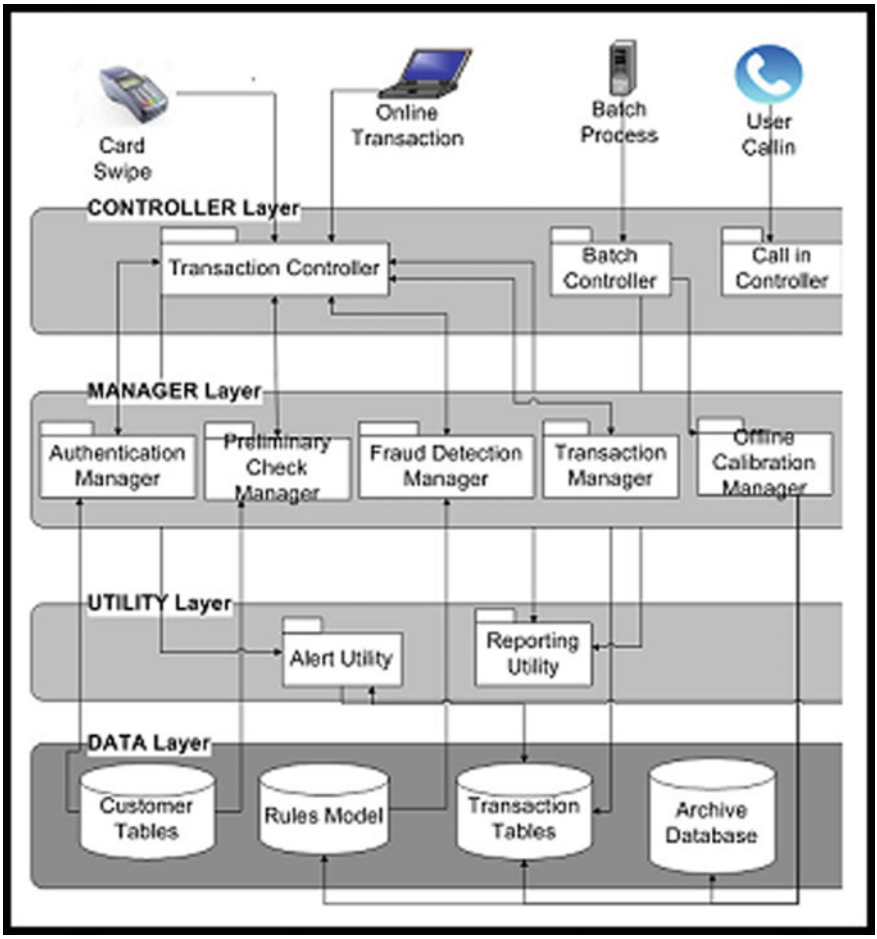

Fig. 2 System architecture

functionality in PDF format, PDFBOX library is used and APACHE POI library is used to support Excel format. After testing the system, it is found that using the fraud detection system, 80% fraudulent transactions are filtered. Heuristics searches also help to detect fraud at the initial instant.

3.3.1 DB Design

The main tables used for fraud detection are:

1. Customer_Vendor_tbl is used to store the RFM attributes, i.e., Recency, frequency, and monetary value for each customer–vendor pair. For each incoming

transaction, the frequency is checked. If it is monthly, then current transaction date and last transaction date for the corresponding vendor is compared. If the difference is less than a month, then after considering certain threshold, the transaction is put on HOLD. Customer care will reach out to the customer to confirm if the transaction is genuine or not.

2. Account_validation_tbl is storing all the different validations for each account. Attribute name in this table represents attribute like location, amount, etc. attribute value represents set of allowed values separated by a comma. For example, it is comma separated list of cities, where the customer can use the card for the transaction. If new city is encountered then the transaction is put on HOLD.
3. Fraud_log_tbl is used to log every transaction that is blocked. The design allows having different attributes for different accounts.

3.3.2 Algorithm

When a customer performs a transaction, then the request comes to credit card fraud detection system and the following algorithm is executed.

Step 1: Perform preliminary checks given below on the incoming transaction.

- *Balance Check*: The credit card balanced is checked to confirm if the there is sufficient balance is the credit card.
- *Validate customer*: Customer is validated to check if he has paid previous bills.
- *Validate card status*: Card status is verified to check if it is in active state.

Step 2: Perform online fraudulent checks given below:

Min/Max amount check: Using the historical data, threshold values for the account are determined and stored in the database. For the incoming transaction, the transaction amount is compared to check if it falls within the account threshold. If transaction amount falls out of threshold, it is marked as suspicious.

- *Customer–Vendor validation*: Using the historical data, the relationship between customer and vendor is determined with respect to frequency and monetary values. For each customer–vendor pair, the minimum amount transacted, max amount transacted, and frequency of the transaction is calculated using historical data and stored in the database. The incoming transaction is verified against corresponding customer–vendor pair for threshold values and frequency. For example, customer ABC transacts with vendor PQR every month between the amounts 1000 to 3000. If the incoming transaction has customer ABC and vendor PQR with amount 7000, the system marks it as suspicious.
- *Location check*: Using historical data, different locations are identified for each customer. These locations are stored as comma separated list in the database. If the incoming transaction has a location which was never found before, it is marked as suspicious.

Step 3: If anyone of above check fails, then transaction is suspicious and is put on HOLD

All transactions that are marked as suspicious then confirmed with the customer. Below algorithm is executed when confirmation is received from the customer.

Step 1: Notify the customer about fraud transaction. Show the validations status to the customer. If the customer confirms the transaction to be fraudulent, abort the transaction. Mark transaction status as "CANCELLED" and no further action is taken

Step 2: If a customer claims that the transactions are genuine, then proceed with transaction successfully and mark as COMPLETED.

Step 3: After the transaction is executed successfully, update the model as given below.

- Recalculate the account threshold values.
- Recalibrate customer–vendor frequency, amount threshold.
- If there is new location found in the current transaction, then update the list of locations.
  With this updated model, the similar transaction will be executed successfully in future.

Step 4: Recalibrate amount threshold, and location list at every end of the day to consider newly added records.

3.3.3 Project Results

Below are the screenshots shown in Figs. 3 and 4 for transaction status, fraud summary, and action taken on suspicious transactions after receiving feedback from the customer.

3.4 Fraud Detection System Evaluation

The data is downloaded from kaggle.com. The datasets contain transactions made by credit cards in September 2013 by European cardholders. Along with this, synthetic data is also added to the database to test different scenarios. The resulted dataset has 152,706 credit card transactions of 120 cardholders. 151,768 transactions legitimate are and 1026 transactions are fraudulent. Criteria used to evaluate the system is alarm rate, sensitivity, false positive rate, precision, negative predictive value, and accuracy as shown in Table 1 [13–15].

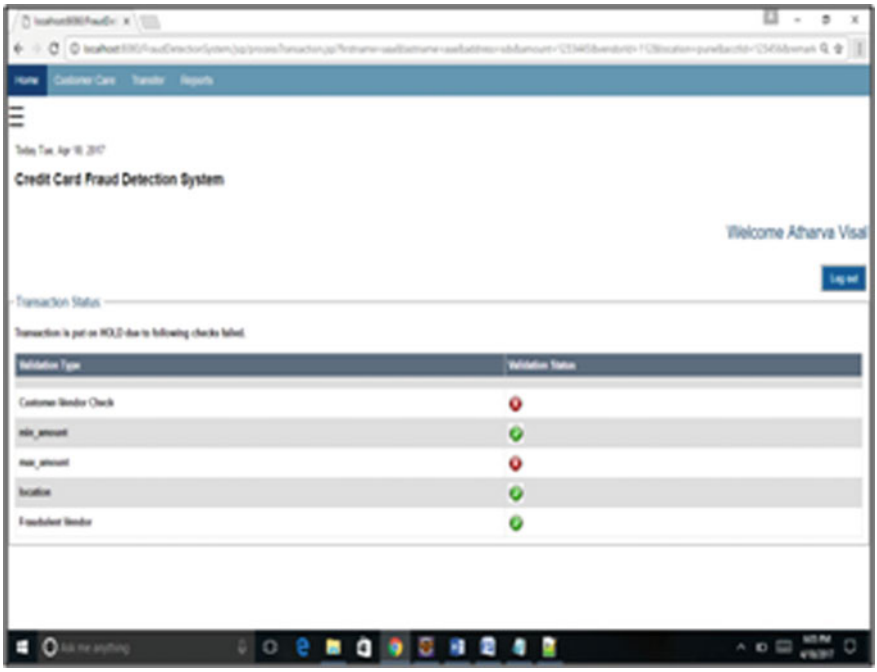

Fig. 3 Transaction status of fraud check

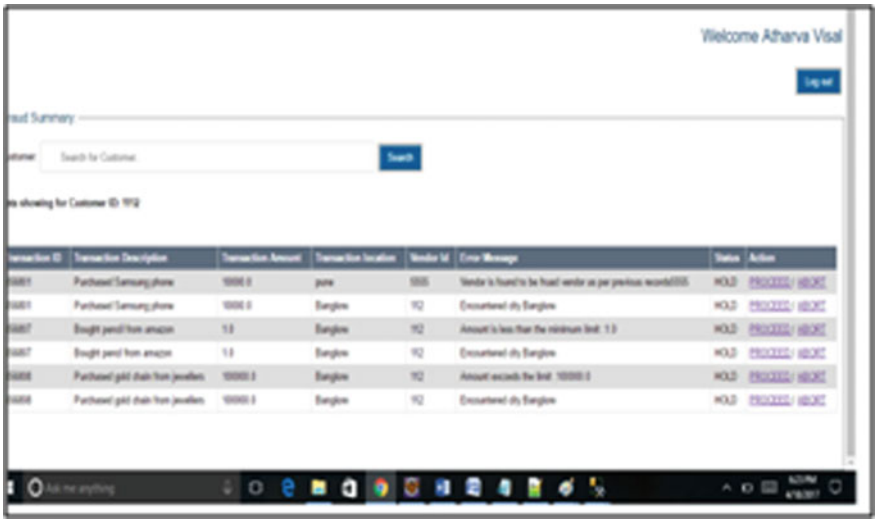

Fig. 4 Fraud summary page

Table 1 Evaluation results

Sensitivity	Precision	Accuracy
43.24%	11.20%	80.12%

4 Conclusion

Credit card fraud detection algorithms have emerged remarkably over last few years. The algorithm used for credit card detection needs to be self-learning and periodically updated. The approach suggested in this paper is most suitable for the changing environment since it changes dynamically. The work is a hybrid system based on Network-based extension and concept drift adaptation. Using past history, the model is determined. The model consists of different conditional attributes. The set of attributes can be different for different customers. Hence, the suggested model is flexible and dynamic with 80% fraudulent transactions filtering rate. Thus, the proposed and developed system is providing the essential backbone to promote the plastic money.

References

1. http://www.pymnts.com/news/2015/global-card-fraud-damages-reach-16b/.
2. Emanuel MinedaCarneiro, "Cluster Analysis and Artificial Neural Networks: A Case Study in Credit Card Fraud Detection," in 2015 IEEE International Conference.
3. Dhiya Al-Jumeily, "Methods and Techniques to Support the Development of Fraud Detection System", IEEE 2015.
4. Mukesh Kumar Mishra, "A Comparative Study of Chebyshev Functional Link Artificial Neural Network, Multi-layer Perceptron and Decision Tree for Credit Card Fraud Detection", 2014 13th International Conference on Information Technology.
5. Andrea Dal Pozzolo, Olivier Caelen," Learned lessons in credit card fraud detection from a practitioner perspective", Expert Systems with Applications 41, 2014.
6. V. Mareeswari, "Prevention of Credit Card Fraud Detection based on HSVM". 2016 IEEE International Conference On Information Communication And Em- bedded System.
7. Carlos A. S. Assis, "A Genetic Programming Approach for Fraud Detection in Electronic Transactions" in Advances in Computing and Communication Engineering (ICACCE), 2015 Second International Conference.
8. Dustin Y. Harvey, "Automated Feature Design for Numeric Sequence Classification by Genetic Programming", IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 19, NO. 4, AUGUST 2015.
9. Kang Fu, Dawei Cheng, Yi Tu, and Liqing Zhang, "Credit Card Fraud Detection Using Convolutional Neural Networks", Neural Information Processing, Springer.
10. Sahin Yusuf, BulkanSerol, DumanEkrem," A Cost-Sensitive Decision Tree Approach for Fraud Detection", Expert Systems with Applications,vol.40, pp. 5916–5923, 2013.
11. Andrea Dal Pozzolo, "Credit Card Fraud Detection and Concept-Drift Adaptation with Delayed Supervised Information".
12. Véronique Van Vlasselaer, "APATE: A novel approach for automated credit card transaction fraud detection using network-based extensions" published in Decision Support Systems 2015.

13. Yiğit Kültür, “A Novel Cardholder Behavior Model for Detecting Credit Card Fraud”, IEEE international conference on commuting and communication engineering, 2015.
14. https://www.creditcards.com/credit-card-news/assets/HowACreditCardIsProcessed.pdf.
15. https://www.kaggle.com/dalpozz/creditcardfraud.