Chapter 26
Computational Techniques for Real-Time Credit Card Fraud Detection

Sangeeta Mittal and Shivani Tyagi

Abstract With e-commerce becoming mainstream and a manifold increase in online transactions, security risks associated with these have become crucial concerns. In this chapter, we focus on the security issues arising out of online credit card usage. Literature in the last two and half decades has been reviewed to analyze the changing attack vectors and solution approaches to this problem. Most common attributes and open datasets of credit card transactions have been compiled to provide a starting point for new researchers. Existing fraud detection methods have been scrutinized for efficacy in addressing key challenges of fraud detection like real-time detection, concept drift, imbalanced datasets, and classifier adaptability. New directions in credit card fraud detection research have also been proposed.

Keywords Credit card fraud · Credit card fraud detection system · Machine learning · Computational models · Classifiers · Supervised and unsupervised learning

1 Introduction

Credit cards have been the main instruments for financial transactions in all online commercial activities since more than two decades. This makes credit card-based payment systems vulnerable to frauds. The history of credit card can be tracked down to 1958 when the first credit card was issued in USA, whereas in India the first credit card was issued in 1981. Since then credit card fraud has incurred losses of billions of credits and is increasing day by day. Credit card fraud is a serious growing problem that occurs as illegal/unauthorized usage of card information, unexpected transaction behavior, or any kind of transaction on an inactive card [1]. According to

S. Mittal (✉) · S. Tyagi
Department of Computer Science and Engineering, Jaypee Institute of Information Technology, Noida, India
e-mail: sangeeta.mittal@jiit.ac.in

B. B. Gupta et al. (eds.), *Handbook of Computer Networks and Cyber Security*,
https://doi.org/10.1007/978-3-030-22277-2_26

the Reserve Bank of India, in January 2018, a total of 36.2 million credit cards were operational. Major e-retailers like Amazon India, Flipkart, and Snapdeal have significantly captured the retail commerce in India. According to Statista portal, in 2018, the percentage of digital buyers has reached to about 60% (https://www.statista.com/statistics/261664/digital-buyer-penetration-in-india/ [Accessed on January 2, 2019]). Thus, the whole ecosystem is conducive to witness a manifold increase in credit card usage in an online transaction. Such usage is called "Card-Not-Present" as instead of physical card, only details of card are required.

The increase in digital payments is also giving rise to a manifold increase in online banking frauds in India. These frauds target banking facilities like credit, debit, and ATM cards, payment gateways, and other net banking techniques. However, a major chunk of frauds are launched on credit cards due to large credit limits offered by banks. An online credit card fraud leaves all the three parties, namely spender, issuing bank, and merchant, in a jiffy and causes economic loss to all of them. Without any specific proactive method being in place in the credit card company, the onus of fraud detection is on the cardholder/card user. The cardholder must report suspicious charges to the issuing bank. The bank then investigates the issue and if evidence of fraud is found then the process for reversing the credit for the transaction is initiated. The cardholder may not seem to be impacted because of fraud in credit card transactions as the issuing bank covers for many losses in fraud scenarios. However, this chargeback is conditional and not applicable in all frauds. Other indirect costs of inconvenience, time to follow-up are also involved. Merchants are also affected from losses due to fraud, particularly in online payments as they must accept full liability.

Even if the fraud loss is borne by the issuing bank, merchants may suffer losses due to unrecoverable costs like shipping cost, card association fees, merchant bank fees, and administrative cost. On the part of the credit card company also, a lot of resources are used in handling the dispute charges.

To address this problem, banks keep on issuing necessary advisories to its users about the secure usage of cards. However, the advisories do not always work against social engineering techniques used by the perpetrators. Thus, in case of an alleged fraud, banks must spend resources in detecting and retracing the source of fraud. The turnaround time for this detection has been several days, which does not prove useful to act as a deterrent against the frauds.

Common approaches suggested for securing smart card-based applications can be applied to credit card fraud detection also [2]. With credit cards issuance becoming easier and rise in buying options, the number of credit card transactions is increasing exponentially. About 130 million credit card transactions with total worth of 1365 crores took place in India in January 2018 (https://www.medianama.com/2018/03/223-india-credit-cards-and-debit-cards-january-2018/ [Accessed January 2, 2019]). Manual inspection of this huge number of transactions to uncover fraudulent ones is an infeasible task. Thus, credit card fraud is a good example of cases where machines can learn from past transactions to tell whether a current transaction is fraudulent or normal. The goal here is to obtain an automated *Fraud Detection System* (*FDS*) to detect all fraudulent transactions without raising a false

alarm. A lot of machine learning-based computational models have been proposed to be used to automate this task [3–32].

In this chapter, computational methods to detect online credit card fraud specifically designed for "CARD-NOT-PRESENT" (CNP) fraud scenarios have been outlined and evaluated.

1.1 Research Contributions

The chapter is an amalgamation of a large body of literature in this area and contributes to the state of the art in the following ways:

1. Define a classification of credit card frauds
2. Outline major challenges in implementing a credit card FDS
3. Summarize the features of datasets used in studies related to credit card frauds and FDS
4. Provides a comprehensive summary of computational techniques proposed for FDS in last two and half decade
5. Critique the existing models with respect to their efficacy in addressing the challenges
6. Methodologically suggest approaches that can improve FDS performance while meeting the challenges

1.2 Chapter Outline

The chapter has been organized in six sections. First section introduces the significance of credit cards in today's commercial scenario. In Sect. 2, type of credit cards frauds and challenges towards designing computational models for fraud detection systems has been discussed. Section 3 discusses credit card datasets and their features. State of the art in categories of computational models proposed to be used for credit card fraud detection has been discussed in Sect. 4. Evaluation of the existing computational model approaches in addressing challenges of FDS has been done in Sect. 5. The chapter is concluded in Sect. 6.

2 Credit Card Frauds and Detection

A credit card is a small plastic card issued by a financial company that authorizes the cardholder to use it for payment of goods and services. The amount of purchase is recorded in the user's account and he has to repay the borrowed sum as well as any other charges agreed upon as understanding between the card company and

the user (https://www.investopedia.com/terms/c/creditcard.asp [Accessed January 2, 2019]; Ways Criminal Steal Money: https://www.gadgetsnow.com/slideshows/15-ways-criminals-steal-money-from-your-debit/credit-card/public-wi-fi/photolist/55414129.cms [Accessed January 2, 2019]). These cards are used by presenting them physically at a Point of Sale (PoS) terminal as well as by furnishing card-specific information during online purchases. An unauthorized use in any of these two forms would be termed as *Credit Card Fraud*.

2.1 Types of Credit Card Frauds

Main motive of credit card fraud is to illegally obtain *physical possession* or *information of card*. However, the modus operandi may differ in various cases. On the basis of instances of frauds that have been discussed in financial information sources, they can be categorized into two main categories described in this section.

2.1.1 Obtaining Physical Cards Illegally

1. *Application Fraud*: Application fraud is when someone obtains a credit card using fake or false information by forging documents and providing fake telephone numbers of residence and place of employment.
2. *Lost and Stolen Card Fraud*: Physical security of credit card is an important factor. If a card is not adequately protected, then it can get accidently lost and fall in the hands of perpetrators. In some cases, an unattended card may be stolen with ill intention. These frauds can be used to launch other frauds.
3. *Counterfeit Cards*: Such frauds are committed through skimming actual credit card information and creating a forged magnetic tape having information about credit card.
4. *Mail Nonreceipt Fraud*: This fraud is also known as "never received issue" or "intercept fraud." It occurs when a user is expecting a new card or a replacement, but a criminal gets its possession before the actual user and starts using it.
5. *Assumed Identity*: All credit card issuance is checked for correct identification of the person to whom the card is being provided. In absence of fool-proof authentication mechanism, a fraudster may impersonate a naive person by obtaining and producing fake address proof and identity document.
6. *Doctored Cards*: One of the ways of fraud is to tamper information of an existing card with the help of a powerful electromagnet.
7. *Fake Cards*: Credit cards may be cloned by copying all the information encoded in magnetic strip and pasting into a new strip to get a fake card. Creation of fake cards can be done by someone who is skilled enough to forge the magnetic strip and the chip and break the complex security and even holograms of real credit cards.

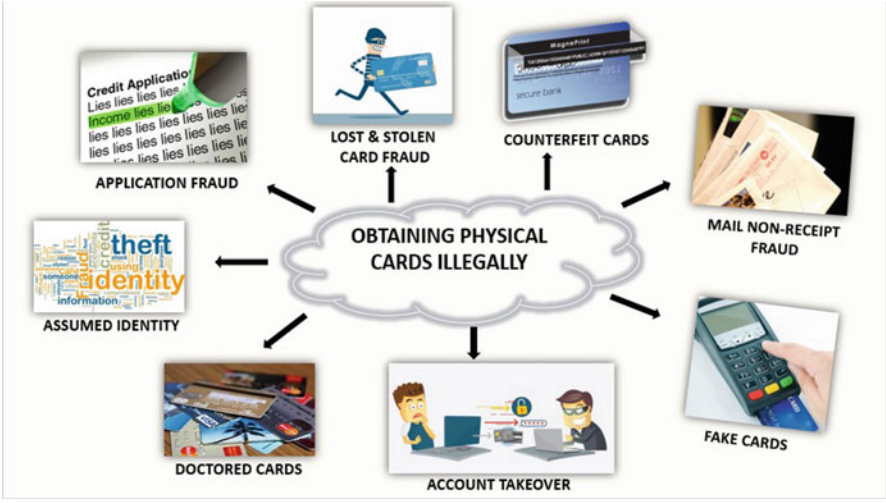

Fig. 26.1 Types of frauds by obtaining credit card illegally

8. *Account Takeover*: Such type of fraud is usually carried out online, where the fraudster talks to the credit card company to replace card by providing relevant documents and information. These attack vectors to physically obtain credit card in an illegal way have been summarized in (Fig. 26.1)

2.1.2 Obtaining Card Information Illegally

Another method to commit credit card fraud is to obtain card information illegally and various methods to do so have been summarized in (Fig. 26.2).

1. *Credit Card Imprints*: Credit card imprints are taken as a measure of security deposit for a service usage like hotel or car rentals. A dishonest service provider or its employee may skim the information, which can be used in fraudulent transactions.
2. *CNP (Card-Not-Present) Fraud*: Card-Not-Present is a type of credit card fraud executed by obtaining card information like a cardholder's name, billing address, account number, three-digit security code, and card expiration date. Such theft of credit card data may occur through online phishing, tampered swipe machines, or shoulder surfing. CNP is generally used in online transactions where the perpetrator does not have to be physically present.
3. *Card ID Theft*: It is the most difficult fraud to detect where the details of credit card become known to a criminal, and this information is used to take over a card account or open a new one. Identity theft constitutes 71% of the most common type of fraud.

Fig. 26.2 Types of frauds by obtaining credit card information illegally

4. *Clean Frauds*: To commit this category of frauds, fraudster does a lot of homework in collecting the user's actual details and working principles of underlying Fraud Detection System. The system does not suspect such a transaction and thus the fraud occurs in a clean manner.
5. *Friendly Fraud*: These frauds are about repudiation. In absence of proper online authentication mechanisms, actual user may deny making a purchase after doing it. The user claims that the card has been stolen before the said transaction.
6. *Triangle Fraud*: As the name suggests, this fraud takes place in three recursive steps. The first step is to create a fake ecommerce store or website that offers popular items at very low price. Users are tempted to make purchases at these sites and their credit card details are stolen. In the second step, goods are purchased from other merchants using previously stolen cards and delivered to the purchaser. The third step is to use the stolen information to make purchases elsewhere. This indirection can help the attack remain hidden for a long time.

First category of frauds, i.e., Illegal physical possession of card requires more resources and physical setup on part of perpetrator and riskier to commit as his/her physical identity can be revealed. These frauds are carried out generally by organized criminal groups. Therefore, these forms of frauds are not very attractive to individual fraudsters.

On the other hand, due to absence of physical identity disclosure, frauds committed by obtaining and misusing credit card information are rosier. With enormous Internet presence of credit card users, obtaining card information has become more feasible. Social engineering attacks as Phishing, Cloned website access due to Pharming attacks, Trojan and backdoor software, malicious insiders, shoulder surfing, and keyboard logging are few vectors by which credit card information can

be allegedly obtained [33]. It can be inferred that information obtained by any of the methods would ultimately be used in online transactions.

From now onwards in this chapter, Card-Not-Present (CNP) will be used as an umbrella term to refer to any of these attacks. Thus, further discussion in this chapter is about second category of attacks.

2.2 Fraud Prevention/Detection System

Frauds aimed at obtaining credit card information can be committed through various vectors discussed in the previous section. These activities occur outside the credit card payment processing systems. An effective Fraud Prevention System (FPS) can contain these by using noncomputational measures like social awareness, proactive network security mechanisms like firewalls, and secure hardware [34]. However, prevention does not always succeed and there are instances of attacks taking place. Thus, the second layer of protection is to detect these frauds as soon as possible [35].

Fraud Detection System (FDS) has been, conventionally, manual where a sampled subset of transactions is audited to check for fraud. This system is neither effective nor scalable. To raise both of these performance parameters, automated computational based FDSs have been designed [36]. Goal of such system is to noninteractively check every transaction, regardless of the presence of prevention mechanisms, for the possibility of being a fraudulent one. Early automated FDSs were simple rule based, where rules were defined by financial experts. Also, these were used on archival data and time to detect was quite high [35].

With volumes of credit card transactions increasing widely, there is a pressing need to detect the fraudulent transaction in real time to prevent losses to the card user, card-issuing company as well as merchant. An ardent requirement is to assess each and every transaction to detect frauds in real time even in presence of dynamic attack vectors.

2.2.1 Heuristics to Identify Fraudulent Transactions

True information about a transaction being fraud can only be generated when the cardholder or the merchant files a complaint with the card-issuing company. To make FDS really effective, its designers use some heuristics to keep an eye on all transactions and raise an alert as soon as a suspicious transaction takes place. The most effective heuristics that can be included in the design model are:

1. A single IP address making multiple simultaneous transactions with different card numbers
2. Multiple IP and e-mail addresses using the same card
3. Large transactions being made than normal amount
4. Identity of user making transaction is not same as the identity of card holder

5. Country of the card usage is different from the country of card issuance
6. Payment made at odd hours according to the local time of the card holder

2.2.2 Challenges in Design of Credit Card Fraud Detection System

Building an effective, real-time, and scalable computation based automated FDS is subjected to several difficulties and challenges enumerated in Fig. 26.3:

1. *Concept Drift*—FDS targeting anomalous behavior suffer from the fact that in real world, profile of normal and fraudulent behavior changes with time. For computational techniques, this leads to a non-stationarity effect in modeling relation between dependent and target variables.
2. *Class Imbalance*—Credit card transactions data are a typical case of highly imbalanced data. In per unit of time, a large number of credit card transactions take place and most of them are genuine. Typically, out of each 10,000 transactions, only 1 has been found to be fraudulent. Traditional computational methods perform poorly in recognizing instances of rarely occurring class, which is actually the class of interest in FDS [37].
3. *Lack of Real-Time FDS*—Most of the existing FDS reported in literature work on archival data that can be used to drive future security policies and forensics. This analysis is effective in a limited manner to detect and block fraudulent transactions in real time [38].
4. *Fraud Detection Cost Overheads*—Many related studies conveniently ignore the overheads in implementing FDS. Cost is however important consideration while estimating the effectiveness of any solution.
5. *Lack of Domain-Specific Metrics*—Existing models have been evaluated on the basis of standard classifier metrics. No standard domain-specific metrics are available to particularly benchmark the performance of credit card FDS.
6. *Lack of Adaptability*—Behavior analysis-based fraud detection methods define normal behavior from past legitimate transactions of a user. Many a time user behavior may evolve due to external factors like family conditions, an increase or decrease in income, and frequent travelling. Existing supervised and unsupervised approaches used in fraud detection systems are not adaptive to changing datasets. Thus, efficiency of detecting new patterns of normal and fraudulent behaviors becomes difficult [39].
7. *Lack of Availability of Know-How*—Existing fraud detection methods are not made public due to apprehension of them being lesser effective. Thus, everyone has to re-invent the wheel and existing knowledge cannot be leveraged.
8. *Unavailability of Datasets*—Credit card companies do not release their labeled datasets for public scrutiny. Many computational methods are based on learning from datasets. Even a few datasets that are publicly available are actually a processed form of actual datasets to hide real variables and their relations.
9. *Lack of Aggregation Possibility to Leverage Cross User Data*—Optimally leveraging transaction data across card-issuing companies and types of cardholders is not possible due to lack of trust among card-issuing companies [28].

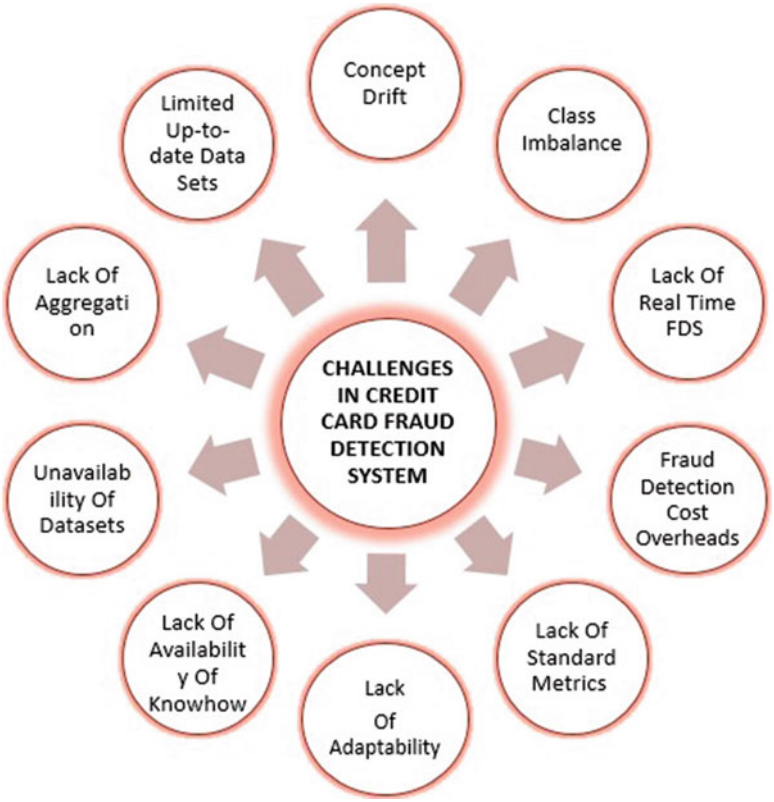

Fig. 26.3 Challenges in credit card fraud detection system

10. *Limited Up-to-Date Supervised Transaction Sets*—Computational models do not have at their disposal recent supervised transactions, provided in the form of investigators' feedback, to dynamically evolve learning models [37].

Challenges 1–6 can be handled computationally, but the remaining ones can be handled only by policy change and collaborative commitment towards fraud prevention. Thus, an effective computational FDS needs to meet these challenges. Particularly, first four are fundamental to the problems of modern-day credit card frauds. An ideal FDS should look for change in transaction patterns that are indicative of fraudulent transaction and produce a suspicion score about it. The score represents possibility of that transaction being fraudulent. If the score is available before the transaction is committed to databases, then it can save a lot of costs to all three stakeholders, namely user, credit card company, and merchant. It will lead to reduced human intervention as only probable frauds would be checked manually. Before discussing the computational models that have been proposed for designing FDS, we first elaborate on the data available in each credit card transaction and archive.

3 Information Available for Credit Card Fraud Detection

An automated FDS would be based on effective computational models. Each such model irrespective of its working principle requires a lot of domain data. In this section, we discuss the features of information that can be available to the models. The feature list has been compiled from the literature on FDS.

3.1 Labeled Credit Card Transaction Datasets

Information available as transaction datasets is input to computational models to solve the problem of fraud detection. A tabular representation of few such datasets and their cardinality has been compiled in Table 26.1. The datasets have been given Ids for ease of reference in further sections. In the dataset description column, the year of creation of data has also been mentioned. In few papers, actual period of data collection was not given. For such works, time of creation of data has been assumed to be some time prior to the publication of paper. Dataset cardinality has

Table 26.1 Summary of some datasets used in research

Dataset ID	Dataset description	Dataset cardinality
D1	European Card Holder Data (2013–2014) [37, 40–44]	284,807 transactions, 28 attributes
D2	Mellon Bank Credit Card Issuer Data (1990) [3]	1,100,000 transactions, 50 attributes
D3	Chase Bank & Union Bank Data (1995–1996) [6]	500,000 transactions
D4	US Bank Data (2000–2001) [14]	25,000 credit card records, 38 attributes
D5	Financial Institute Data (Webbiz-Ireland) (2004–2008) [16]	4 million transactions, 23 attributes
D6	Large European card-processing company Data (2012) [17]	80,000,000 individual transactions, 27 attributes
D7	Actual Fraud Transactions combined with the different number of normal transactions (before 2012) [18]	42 attributes with imbalanced ratios as 236, 23.6, and 4.7
D8	Australian Bank Data (2003) [20]	640,361 total transactions of 21,746 credit cards
D9	Vesta Corporation Data (before 2012) [21]	206,541 transactions
D10	Spanish Bank Dataset (2011–2012) [22]	180 million transactions, 10 attributes
D11	Major Financial Institution Data (before 2017) [24]	86 million transactions, 69 attributes
D12	E-tail Data (Jan 2015–Aug 2015) [26]	347,572 transactions, 70 attributes
D13	Retail Banking Data (before 2018) [30]	80 million transactions, 5 attributes
D14	Universo Online Inc. Data (2014) [31]	903,801 transactions

been mentioned to state the amount of information that has been used to study the problem of frauds in credit card-based payments. These datasets are highly imbalanced in ratios of fraudulent versus nonfraudulent transactions.

Most of the datasets have been obtained by the researchers from their industrial partners and, due to confidentiality commitment, have not been provided publicly. This restricts the usage of datasets in further researches as well as verification of existing results. Only Dataset "D1" is publicly available and is highlighted in bold. It is a publicly available and processed, and real dataset is available for free download at [45]. The dataset contains total 284,807 transactions made in September 2013 by cardholders of a European country. Out of these, only 492 transactions are fraud, which makes it highly imbalanced. The data have been made available as 28 principal components computed out of actual data, owing to confidentiality issues. Apart from that, there is "Time" attribute, which is the time elapsed since first transaction in the dataset. "Amount" attribute contains the sum of money involved in the transaction. This feature can be used to compute cost of an undetected fraud. Data have been labeled as "1" in case of fraud and "0" otherwise.

All the datasets mentioned in Table 26.1 have different credit card usage information attributes. On careful examination, these can be divided into three categories, namely customer's basic information, current transaction descriptors, and user-specific usage history.

1. Customer basic information descriptors

   (a) Whether card holder is male/female
   (b) Card holder's age—behavior of aged customers is more predictable
   (c) User identification information in terms of associated account number
   (d) Identification number of card, generally a 16-digit number
   (e) Type of card: Master, Visa, etc.
   (f) Encrypted customer ID: customer identifier
   (g) Date of birth
   (h) Registration date and time: the date and time the customer registered to open their account
   (i) Country of residence of the card holder

2. Current transaction descriptors.

   (a) Category of merchant assigned as code by card-processing company; for example, jewelers, electronics, etc.
   (b) Payment ID
   (c) Status of whether the transaction was successful or declined
   (d) Place of transaction
   (e) Currency
   (f) Quantity of current product ordered
   (g) Category of goods being purchased
   (h) Brand of goods
   (i) Is shipping country the card country
   (j) Payee account number

Table 26.2 Common attributes

S. no.	Attribute name	Description	Data set including attributes
1.	Transaction amount	Amount of money spent	D1–D3, D5–D6, D8, D10–D12
2.	Merchant code	Encoding categories of sold goods	D2, D6, D8, D11
3.	Transaction date	Date at which the transaction was performed	D2, D5–D6, D8
4.	Transaction time	Time at which the transaction was performed	D1–D2, D6, D10, D12
5.	Transaction place/recipient address	Geographical location of transaction determined by IP address	D5
6.	Fraud	1 if the transaction has been recognized as a fraud, 0 otherwise	D1, D6, D8, D10–D12
7.	Credit card number	The 16-digit unique credit card number	D8
8.	Current balance	Account balance after transaction	D5, D11
9.	Transaction type	Type of transaction: purchase/payment/transfer to other account	D5, D12
10.	Purpose	Car, real estate, life insurance, property, etc.	D2, D6, D12

(k) Transaction amount
(l) Country where the transaction took place
(m) Number of transactions in the last 48 h
(n) Accumulated amount of transactions in the last 48 h
(o) Number of terminals used in the last 48 h

3. History descriptors

(a) The payments made to the account in recent times
(b) Fraud rate: average rate of illegal operations, for all cards, in the last 50,000 transactions

Many of these attributes may not be directly available in each transaction but can be derived from other existing values. For example, Bahnsen et al. [17] derived 260 attributes from selected 14 original attributes in dataset. History and customer basic information descriptors are used to work out behavior-based fraud detection while current transaction-based descriptors can be used to find misuse [24]. On examining the body of work, it was found that the databases considered for this problem somewhat vary in the type of information considered to be available for designing the computation models. The common transaction attributes that were used in most of the datasets for credit card fraud detection have been presented in Table 26.2. Name of the datasets where the attributes were available for learning the models has also been mentioned in the last column.

Data types of the attributes are mixed and range from numerical to categorical and ordinal. For example, transaction size and current balance are real-valued and merchant code and country names symbolic.

4 Computational Techniques for Fraud Detection

Computational models proposed for credit card fraud detection date as long as the frauds themselves. A variety of statistical, machine learning and data mining tools have been utilized to obtain an automated fraud detection system in the presence of mixed data types. Table 26.3 summarizes five yearly snapshots of main techniques used against this problem from 1994 to till now.

Currently, the techniques used for credit card fraud detection can be classified into the following categories:

- *Fraud Analysis*: Deals with supervised learning for identifying misuse detection
- *User Behavior Analysis*: Deals with unsupervised learning for anomaly detection

If a large number of labeled transactions are available, then machine learning-based classifiers can be trained to distinguish future fraudulent and normal transactions. These classifiers use label information to model the two types of transactions. Various supervised learning methods like decision trees (DTs), back propagation neural networks, support vector machines (SVMs), random forests, and Bayesian networks (BNs) have been applied to obtain the desired result of detection [3–6, 14, 17, 30, 31, 37, 41]. They are effective only for detecting frauds following similar patterns as those identified as fraud in past.

However, these methods are unsuitable for recognizing new patterns of fraudulent transactions. Unsupervised class of methods is agile in adapting to novel frauds and thus can be used against an adaptive fraudster. Self-organizing maps, peer group analysis, break-point analysis, and competitive learning are few unsupervised methods applied for detection of frauds [8, 15, 21, 24, 46].

Another class of methods detects fraud according to individual behavior analysis of individual user, which was ignored in machine learning methods. These involve learning profile of normal transaction pattern for each user based on her or his normal transactions. Profile of current transaction is matched against this profile and a suspicion degree is assigned to each transaction based on the user's profile [22, 27, 44].

Pre-processing Transaction attributes are of mixed data type including categorical, ordinal, binary, numeric, and string. Binning, averaging, normalization, ordinal to numeric, categorical features to numeric, ranking, and ordering are few pre-processing methods applied to map input variables to a set of more descriptive features [5]. Pre-processing of data is a required step before applying many computational models based on machine learning.

Table 26.3 Summary of credit card fraud detection research using closed datasets

Study	Method used	Dataset_ID/description
[3] Credit Card Fraud Detection with a Neural-Network (1994)	Artificial Neural Network (ANN)	D2: 1,100,000 transactions over 2 months' period. Fifty attributes were mapped to 20 and used as input to the ANN
[4] Neural Networks Compared to Statistical Technique (1995)	Discriminant Model analysis	6-months data of real accounts with more than 50 million transactions
[5] Density-Based Clustering and Radial Basis Function Modeling to Generate Credit Card Fraud Scores (1996)	Radial Basis Function Network (RBFN) with Density-based clustering	Real data with 37 attributes
[6] Distributed Data Mining in Credit Card Fraud Detection (1999)	AdaCost algorithm	D3: a set of 20% fraud and 80% nonfraud transactions from Chase Bank
[7] Unsupervised Profiling Method for Fraud Detection (2001)	Behavioral outlier detection techniques used Peer Group Analysis and Break Point Analysis.	Per week spending data of 858 accounts over a period of 52 weeks
[9] Parallel Granular Neural Networks for Fast Credit Card Detection (2002)	Parallel Granular Neural Network (GNN)	Real transaction datasets, details not provided
[15] Real-Time Credit Card Fraud Detection Using Computational Intelligence (2007)	Self-organizing map-based clustering for Neural Network, Similarity functions: Euclidean-distance and Gravity function	Test database extracted from an actual banking database
[16] Identifying Online Credit Card Fraud Using AIS algorithm (2010)	Artificial Immune System (AIS) benchmarked against logistic regression model	D5: 4 million transactions from 462,279 different customers with 5417 fraudulent cases
[17] Cost Sensitive Credit Card Fraud Detection Using Bayes Minimum Risk (2013)	Cost-sensitive method based on Bayes minimum risk	D6: A set of 80,000,000 transactions with 27 attributes. Fourteen attributes were used. Ratio of frauds was 0.025%
[18] Detecting Credit Card Fraud by Genetic Algorithm and Scatter Search (2011)	Genetic Algorithm and Scatter Search	D7: Custom-defined datasets varying imbalance from 4.7 to 236
[20] Artificial Immune Systems for The Detection of Credit Card Fraud (2012)	Artificial Immune System (AIS)	D8: 640,361 total transactions, with 21,746 credit cards
[21] Improved Competitive Learning Neural Networks for Network Intrusion and Fraud Detection (2012)	Iterated Competitive Learning Network (ICLN) and Supervised Iterated Competitive Learning Network (SICLN)	D9: 206,541 transactions: 204,078 transactions are normal and 2463 are fraudulent

[41] Credit Card Fraud Detection and Concept-Drift Adaptation with Delayed Supervised Information (2015)	Along with Balanced Random Forest (BRF), two traditional learning approaches for FDSs, namely, • classifier is retrained on daily data of most recent supervised samples • an evolving ensemble approach where older results are replaced by better ones	D1: 284,807 transactions, 28 attributes. Dataset was available from 2 years: 2013 dataset with 160k transactions and 304 frauds; 2014 dataset with 173k transactions and 380 frauds
[24] Adversarial Learning in Credit Card Fraud Detection (2017)	SMOTE for oversampling and Gaussian Mixture Models for classification	D11: 36 GB of data consisting of 86 million anonymized transactions. Eleven relevant attributes out of 69 actual were used
[27] A New Credit Card Fraud Detecting Method Based on Behavior Certificate (2018)	Behavior certificate (BC), which reflects cardholders' transaction habits	Synthetic data: normally distributed with mean = 275 and standard deviation = 20; 6-month period, amount is normally distributed for 10 cardholders, out of them for 5 cardholders, mean = 50 and standard deviation = 10. For other 5 ones, mean = 1000 and standard deviation = 150
[28] A Utilitarian Approach to Adversarial Learning in Credit Card Fraud Detection (2018)	Feed-forward Adversarial Learning Game Algorithm	8,000,000 transactions with 0.01% fraud cases
[30] Deep Learning Detecting Fraud in Credit Card Transaction (2018)	Deep learning or General Artificial Neural Network with built-in time and memory components such as Long Short-Term Memory	D13: 80 million transactions; 5 features were used
[31] A customized classification algorithm for credit card fraud detection (2018)	Fraud-BNC, a customized Bayesian Network Classifier (BNC)	D14: 903,801 transactions with 1.8% fraudulent ones described by 424 attributes
[22] Credit Card Fraud Detection through Parenclitic Network Analysis (2018)	Features for MLP-based classifier learnt from parenclitic network of transactions	D10: 15 million operations realized by 7 million cards, for a total of 250 GB of information
[37] Credit Card Fraud Detection: A Realistic Modeling and A Novel Learning Strategy (2018)	Proposed learning strategy using alert-feedback interaction	D1: 284,807 transactions, 28 features, frauds account for about 0.2% of all transactions
[44] Transaction Fraud Detection Based on Total Order Relation and Behavior Diversity (2018)	Logical graph of BP (LGBP)	D1: 284,807 transactions, 28 features
[46] Credit Card Fraud Detection Using Self-Organizing Maps (2018)	Self-Organizing Maps (SOMs)	Synthetic data simulated according to the Ukrainian credit card market

4.1 Computational Models Based on Supervised Machine Learning

This section summarizes the knowledge gathered from literature on the use of supervised methods for credit card fraud detection.

- *Discriminant Analysis*—In discriminant analysis, a set of independent features is selected to learn a model or mathematical equation to classify given data into two mutually exhaustive classes. In [4], fraudulent and nonfraudulent transactions were used to learn a statistical model of discriminant analysis to label good and bad accounts. The model, when run over more than 50 million transactions, gave good results of 4% false positives and 85% accuracy.
- *Decision Trees and Random Trees*—Decision Tree (DT) is a method of supervised classification in which root node is created first for one of the attributes. The node is split further according to all possible values of root attributes. This process of creating new nodes is repeated until a stopping criterion is met. All leaf nodes are associated with class labels to which most of the samples terminating to that leaf belongs to. Random tree is a decision tree that uses a random subset of attributes to create decision tree classifier. The subset size is defined using a subset ratio parameter [43].
- *Radial Basis Function Networks (RBFN)*—RBF model is learnt in two phases. Training includes learning cluster centers and scaling parameters. Centers can also be computed by vector quantization or tree classification algorithms. In the second phase, weights are computed according to cluster centers. One of the advantages of two-phase learning in RBF networks is the possibility of using unlabeled training data in the first phase. In [5], RBFN-based model has been trained to classify transactions as fraud and nonfraud. The results were claimed to be better than ANN with back propagation.
- *Meta-Classifier*—Meta-classifier, also known as ensemble learning, achieves strong classification results by combining results of multiple classifiers where each of the chosen classifier may be individually weak. In [6], AdaBoost learning was modified considering domain-specific misclassification cost as the main decision parameter. The algorithm has been named as AdaCost by the authors. Four base classifiers, namely C4.5, Classification and Regression Trees (CART), Ripper, and Bayes, were used to create the ensemble using class-combiner strategy. Authors were able to obtain 3% reduction in cumulative costs as compared to AdaBoost on Dataset D3.
- *Bayes Minimum Risk Classifier*: This classifier considers trade-offs between probability of a data sample falling into one class and cost associated with classification. In [17], Bayes minimum risk classifier was used and evaluated on cost-to-fraud metric suggested by them. It was seen that a 23% more saving could be obtained because of this method as compared to state of the art available then.

- *Random Forest*—Random forest is a result of applying a number of random tree-based classifiers and applying majority voting to determine classification result. Dal Pozzolo et al. [41] used balanced random forests of 100 trees to solve the problem of concept drift. Classifiers were updated on recent and delayed feedbacks. It was found that better results are obtained when recent feedbacks are given more weights than delayed ones.
- *Bayesian Network Classifier*—A Bayesian Network (BN) classification approach involves learning a Bayesian network of interdependencies between various independent attributes and probabilities to quantify each network link's strength in conveying the relation as conditional probability tables (CPT) for each node. Learning a BN is about learning the network structure as well as the probabilities. Authors in [31] applied BN-based classification approach to detect frauds in D14 database. For learning the network structure, they have suggested an evolutionary algorithm and named the obtained classifier as Fraud-BNC. Using this classifier, 98.31% of nonfraudulent transactions correctly classified against 71.87% with another classifier.
- *Artificial Neural Networks (ANN)*—ANNs are also known as Feed-Forward Neural Networks that use back propagation algorithm for training purpose. The connections between the units are acyclic. Information in terms of weights at each layer acts as input to next layer nodes. All intermediate layers, that is, those apart from input and output layer, remain hidden [43].

  In [3], neural networks were trained on labeled dataset and obtained model was used in real implementation in Mellon bank. The model was run every 2 h to verify transactions committed since the last commitment. Thus, making the time to detect was more than 2 h.
- *Deep Learning (DL)*—DL is recent popular concept based on the concept of a multilayer perceptron network. In this type of classification, each layer learns weight such as to reproduce the output itself. Stochastic gradient descent is the error function used to decide the direction to move in the state space. Various nonlinear activation functions like maxout activation, rectifier, and tanh have been used for approximation of complex functions. In [30], various deep learning models namely Artificial Neural Networks (ANNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTMs), and Gated Recurrent Units (GRUs) were used to solve FD problem. Six hidden layer GRU architecture with 150 nodes produced the best performing model giving an accuracy score of 0.916.
- *Decision Tree-Based Classifiers*

  – *ID3*—Iterative Dichotomiser 3, popularly known as ID3, is the basic decision tree algorithm. At every step of tree creation, entropy of all remaining attributes is computed and one with minimum entropy is chosen [29]. Further nodes are creating by creating subsets of data values of chosen attribute. One of the drawbacks of ID3 is that it is disoriented because of presence of attributes with a large number of possible values [47].

- *C4.5*—Ross Quinlan, designer of ID3, extended his algorithm to C4.5 algorithm for serious commercial usage. To overcome the limitation of ID3, the stopping criterion used in C4.5 is normalized information gain instead of entropy. The attribute with the highest normalized information gain is chosen to make the decision [47].
- *C5.0*—It is immediate successor of C4.5 by the same designer. C5.0 improves its predecessor in many ways. It fares better in terms of speed, memory usage during runtime, lesser storage, and search complexity due to smaller trees. It also supports pre-classification tasks like winnowing, weighting, and postclassification like boosting.
- *CART*—Classification and Regression Trees (CART) is a type of decision tree. The tree is flexible such that if variables are categorical then output is classification and if the input is real valued output is real valued, that is, regression. Like basic decision trees, nodes are split on rules based on values of features. This splitting is recursively repeated until stopping criteria like no further information gain is achieved. For choosing attribute to split, the strategy is to choose the attribute whose gini Index is least after splitting [47].

Authors in [37] compared decision tree-based methods with computationally expensive Support Vector Machine (SVM)-based approaches. Results claim that decision tree methods are able to provide best 89% testing accuracy as compared to 83% by SVM.

- *Hybrid Supervised Approaches*—In these types of systems, accurate fraud detection has been obtained by applying multiple approaches in different phases. Authors in [14] apply three-stage FDS. In the first stage a rule-based filter is used to flag a suspicious transaction. This transaction is given a score on the basis of Dempster Shafer's theory of evidence. In the third stage, Bayesian learner that has been computed from transaction history database of user has been utilized to update the value of evidence and eventually term the transaction as fraud or nonfraud.

Supervised algorithms work by learning from true labels. But they need large training datasets [39]. ANNs give satisfactory classification rate even for large transaction datasets.

4.2 Computational Models Based on Unsupervised Machine Learning

Unsupervised learning is useful in studies that need to detect changes in behavior or unusual transactions. Actual labeled fraudulent and normal transactions are not available. An initial set of transactions considered as normal is used to start the classification process. Further transactions with any significant deviations from this set are considered to be fraudulent. Unsupervised techniques can be used to model

user similarity. Some methods for unsupervised fraud detection in credit data have been explained below.

- *Peer Group Analysis (PGA)*: PGA is an unsupervised learning approach. It is about finding a set of peers by grouping similar objects over a time window and then calculating peer group statistics. Any object deviating from its peer group behavior is pointed out as suspicious. In [5], peer group analysis was used to form peer groups of fraudulent and nonfraudulent transactions. Authors propose to modify the method by changing the length of the time window used to determine the peer group. This change in window size caters to the need to detect short-term changes in spending behavior [8].
- *Break-Point Analysis*: In a set of observations, a break point is an observation or point of time where anomalous behavior is detected. As against supervised approaches that worked towards generalization of a normal versus fraudulent transaction, this method works for individual users. A transaction can be abnormal for one user but perfectly normal for any other user. Thus user-specific break-point analysis tracks its anomalous behavior and generates alarm when a break point is reached. In [8], BPA has been used to identify changes in spending behavior based on the transaction information of an account.
- *Self-Organizing Maps (SOM)*: It is an unsupervised learning method that configures the underlying neural network according to the topological structure of the input data. Weights of neurons are iteratively tuned to approximate the input data. Clustering method of SOM has been found to be appropriate for analyzing deviation in customer behavior in [46]. Self-organization is used to learn patterns from existing unlabeled transactions and keep them in different clusters according to similarity in patterns. Eventually two clusters representing legal card holder's and fraudster's behavior are found. Authors in [15] use SOM-based clustering to identify certain "suspicious" transactions that require further review.
- *Improved Competitive Learning Network (ICLN)*—It is type of neural network. Authors in [21] used unsupervised methods to learn natural clusters within the data. On applying their method on dataset D9, recall rate was only 57.4%. This metric was improved significantly when ICLN was modified to supervised ICLN, changing the recall rate to 79.1%.
- *Adversarial Learning*—The concept of adversarial learning is a specialized area of machine learning that learns the dynamic adversary behavior model and updates the classifier to adapt to the changed behavior. Modeling of adversarial scenarios makes varying assumptions on the amount of knowledge the adversary has about the classification system. Zeager et al. [24] investigated the method to model adversary's optimal strategy and update a logistic regression classifier, assuming that adversary can compare different strategies used by FDS. Gaussian Mixture Models (GMM) has been chosen as an unsupervised way to create three distinct strategies that the adversary can choose from [24]. Each transaction is assigned to the strategy it most likely belongs to. Best strategy is the one that gives adversary the highest false-negative rate. The classifier is retrained

to counter this strategy. In this solution, the classifier was being checked for retraining in every game between adversary and FDS. Authors in [28] improved upon this approach by optimizing the retraining decision. They incorporated economic value into the decision to retrain and the selection of strategy. Results indicate that performance similar to [24] is obtained even when training is manifold reduced.

One of the advantages of using unsupervised neural networks over similar techniques is that these methods can learn from data stream. The more data passed to an SOM model, the more adaptation and improvement on result are obtained. More specifically, the SOM adapts its model as time passes. As a result, the fraudulent use of a card can be detected fast and effectively. However, neural networks have some drawbacks and difficulties that are mainly related to specifying suitable architecture on the one hand and excessive training required for reaching to best performance on the other hand.

4.3 Computational Models Based on Nature-Inspired/Biologically Inspired Computing

- *Genetic Algorithms (GA)*—Genetic algorithms are evolutionary algorithms that aim at obtaining better solutions as time progresses. The initial population selection process is random, which limits the probability of goodness of initial population to the random function chosen. In [18], genetic algorithm has been utilized to solve fraud detection problem. However, the initial population is selected by scatter search method. This method involves improving the random selection local improvement method like mutation. The algorithm works with 1050 fraud transaction and created three databases by varying number of normal transactions. The study concluded in finding the best set of variables that determine the transaction to be fraud or nonfraud.
- *AIS*—Artificial Immune Systems (AIS) are a recent branch of artificial intelligence based on the biological metaphor of the human immune system [48]. The immune system can distinguish between self and nonself, or more appropriately, between harmful nonself and everything else. Thus, AIS-based mathematical model assumes everything it has not seen as non-self. AIS can thus be constructed to flag "nonstandard" transactions without having seen examples of all possible such transactions during training of the algorithm. The technique has been used for identification of anomalous credit card transactions [16]. Authors implemented AIS to find frauds in Dataset D5. According to results, Artificial Immune System could achieve up to 98.96% accuracy. The misclassification of a large number of normal transactions makes this algorithm unsuited for fully automatic operation. However, potentially fraudulent transactions could be subjected to further automatic or human processing to reduce the number of

false negatives (FN). Authors in [20] also applied AIS to all types of credit card frauds namely lost, stolen, skimmed, and mail/phone fraud and achieved average detection accuracy up to 71% approximately.

4.4 Computational Models Based on Other Miscellaneous Approaches

Recently many other approaches have been explored as solution to the problem [22, 27, 44]. A hybrid data mining/complex network classification algorithm has been proposed in [22]. Complex networks were used to synthesize complex features from transaction logs. Specifically, parenclitic networks, a network reconstruction technique that works towards finding difference between a given data instance and a set of training instances, has been utilized. Network structure formation is based on finding topological features whose correlation strongly differentiates normal and abnormal transactions. About 5.9% increase in the Area Under the Curve (AUC) was observed when networks were trained with an objective of minimization of false positives.

A fraud detection method based on Behavior Certificate (BC) has been proposed in [27]. BC certifies the user's general as well as special case (festival/weekends) behavior features that FDS can verify. From the set of behavior features in transactions datasets, a binary behavior feature vector of 13 values, namely (1) "Weekday" (2) "Weekend" (3) "Festival" (4) "Normal Day" (5–8) "Interval$_i$" ($i = 1$–4) are four time-intervals (9) "Location"—area code (10–13) "Range$_i$" ($i = 1$–4) as transaction amount ranges. Interval and range values are user specific and learnt from his or her past spending behavior. On every new transaction, a risk value is computed on the basis of the cardholder's BC and an alert is generated if the risk is above threshold. The method performed well on a synthetic simulated dataset and gave specificity values of up to 92%.

Generally, the attributes of a transaction are totally ordered. For example, transaction_time → transaction_location → category_of_good → amount → shipping_address. Based on the total order relation and the transaction log of a user, we can construct a logic graph of BP (LGBP) for the user, which represents the dependent relations of all attribute values of this user's records and covers all transaction records [44]. In LGBP, path-based transition probability is computed. Based on this probability, recognization degree for a given transaction record and BP of user is calculated. Recognization degree of a transaction represents the probability of the transaction in the history even on considering user's transaction diversity. It has been reported that mean recall is about 95% and mean precision is about 85% on self-accumulated datasets. User's BP can be updated by event- or time-driven policies.

Telecommunication, computer intrusion, and money laundering share computational techniques for credit card detection [10, 12]. Fraudsters are adaptive to the

protection mechanism in place. AIS and adversarial learning are tuned towards learning changing attacker profile. There is a dearth of published literature on fraud detection. Machine learning techniques based on supervised neural networks dominated the commercial fraud detection systems in the late nineties [10]. Outlier detection methods for behavioral outliers are an interesting line of approach in absence of a lot of labeled data [12]. Algorithms for adaptive pattern recognition and statistical modeling combined with rule-based expert systems also gave promising models [12]. With evolution in computational models in the first decade of the twenty-first century, other techniques for fraud detection were also explored. This included Hidden Markov Models, rule-induction techniques, fuzzy system, decision trees, Support Vector Machines (SVM), and K-Nearest Neighbor algorithms [19]. Among supervised methods, K2, TAN, Naïve Bayes, Logistic regression, and J48 decision trees were tried by few researchers [25]. A comparison between performance of logistic regression, random forests, and support vector machines was carried in [23]. Random Forest proved to be the most effective with highest 93.5 AUC. A study on hybrid methods that use AdaBoost and majority voting methods has been done in [43]. It was found that on single classifiers accuracy of fraud detection rates vary from 7.4% for Linear Regression (LIR) up to 100% for random forests, gradient-boosted trees, decision stump, neural network, multilayer perceptron, and logistic regression. AdaBoosting improved performance of naive Bayes, decision tree, and random tree. In LIR, the improvement was drastic from 7.4 to 94.1% fraud detection accuracy. Majority voting method further improved the results giving 95–100% fraud accuracy. A deep learning-based fraud detection model has been implemented in [33] and gives a good area under curve for preserving privacy of card parameters. In [49], a method to detect fraudulent transactions has been given by first shortlisting suspicious transactions using fuzzy c-means algorithm and based upon suspicion score a neural network-based classifier labels the transaction as fraud or nonfraud.

Despite having high accuracy, most of the machine learning methods suffer from high false-positive rates resulting in a nonreliable system, as too many resources are wasted verifying legitimate transactions instead of identifying anomalous ones.

5 Evaluation

In Sect. 2, major challenges faced by effective FDS were enumerated. In the previous section, many computational models were discussed with respect to their application as classifiers in fraud detection system. In this section, the extent to which the models are able to address the challenges has been analyzed.

5.1 Handling Class Imbalance

One of the biggest swinging blocks is the immense data and its distribution [50]. In almost all datasets, fraudulent transactions were significantly lower than normal healthy transactions accounting to around 1–2% of the total number of observations. The algorithms used for credit card fraud detection tend to produce unsatisfactory classifiers when faced with imbalanced datasets. The common methods used for dealing with unbalanced classification are:

- *Under sampling*—For large transaction datasets, some legitimate instances can be dropped to create a balanced dataset. This process of selectively choosing majority class instances is called under sampling [17] used under sampling approach to handle skewed class distribution. Five different databases S1, S5, S10, S20, and S50, each one having a different percentage of frauds 1%, 5%, 10%, 20%, and 50%, respectively, were created. Authors in [27] did undersampling to get 10:1 ratio.
- *Oversampling*—In cases where data are imbalanced as well as small in size, oversampling can be used. It involves synthetic creation of minority class samples sometimes by mere duplication. Bootstrapping and SMOTE (Synthetic Minority Over-Sampling Technique) are some more sophisticated techniques to create new samples [51]. A combination of over- and undersampling is often successful as well [21].
- *Synthetic Minority Over-Sampling Technique (SMOTE)*—It is a method to systematically create new synthetic samples of minority class transactions [52]. Depending upon the amount of oversampling required, neighbors from the k nearest neighbors are randomly chosen and their convex combinations are prepared to obtain new samples. SMOTE has been used for generating artificial fraud transactions for creating balanced datasets [52].
- *Stratified Sampling*—This method involves dividing the dataset based on some characteristics of data population. After dividing the population into the strata, one can randomly select samples from each subset [51].

In [6], with given 20:80 class distribution, four subsets are generated from each month for generating 32 datasets with 50:50 distribution. In [41], delayed samples and windowing were used to handle concept drift problem. In [16], Artificial Immune System (AIS)-based solution does not depend upon knowledge about fraudulent transactions. It learns normalcy from normal transactions and anything that is not normal is termed as fraudulent. Thus, skewness in the class distribution does not affect results. Supervised Incremental Competitive Learning Network (SICLN) performed very well on highly skewed data [21].

5.2 Handling Concept Drift/Adaptability

A fraudster tries to assimilate information about detection strategies by trying dummy attacks. It then tends to change fraud behavior like scale, frequency, and target, over time to avoid getting caught by an FDS. This problem is known as Concept Drift in FDS. The model should be able to detect and respond to it. Even if concept drift is detected early, an FDS still takes time before a new model is trained to use such information [41]. Another nonstationarity that can occur in FDS is due to variable behavior of card users caused by unknown, seasonal, periodical, trend-specific factors. Techniques that work towards adapting the classifier against concept drift adopted by fraudster are more successful in containing the frauds [42].

In practice, concept drift adaptation is achieved by combining ensemble methods and resampling techniques. In [41], concept drift adaptation is achieved by training a classifier over a sliding window and by using ensemble approaches by separating delayed supervised samples from feedback samples.

5.3 Ensuring Real-Time FDS

An ideal FDS aims to detect fraud before the transaction approval process. This real-time requirement is difficult to fulfill given the inherent delay in obtaining information about fraud occurrence due to verification latency [41]. Moreover, a detailed examination of every transaction in real time may not be possible as customers will not like to wait for this process. But during order processing period, fraud detection mechanism can be applied.

Towards this, authors in [15] demonstrated the use of clustering and filtering capabilities of SOM for marking the transactions that deviate away from the customer's cluster of behaviors as "suspicious" [15]. All transactions that are marked suspicious are put on hold and sent for extended authentication process. Other transactions were allowed to proceed without any delay. None of the studies have proved real-time behavior by discussing results of time to detection.

Concept drift and real time are conflicting requirements, as adapting to concept drift requires time. Both cannot be fulfilled by conventional supervised and unsupervised models described earlier.

For achieving a real-time Fraud Detection System computational models related to Streaming Analysis, Spark Streaming and Time Data Analytics can be explored [23]. These techniques will be useful in designing an FDS that learns from transaction streams in an unsupervised manner and thus is adaptive to both changing fraudster strategy and customer behavior.

5.4 Fraud Detection Cost Overheads

Many approaches of robust fraud detection involve a lot of pre-processing and complex model learning. However, in evaluation and performance comparison, these overheads have not been considered. A direct comparison between outcomes of methods without considering the model building effort would be unfair. This aspect has been ignored by all the works in FDS and can be a future line of research. Apart from that, when a transaction is refused, the investigators contact card holder to verify if it is the case of a false alert or a real fraud. This cost in terms of man hour or amount spent should also be considered as a cost-enhancing factor. Sometimes a false alert may lead to card being blocked (for example, as a preventive measure, if customer could not be contacted) then the inability to make transactions can translate into big losses for the customer.

For all these reasons, determining an all-inclusive cost measure is a challenging problem in credit card detection that has not yet been satisfactorily solved till now.

5.5 Lack of Domain-Specific Metrics

Credit card FDS has been seen as classification problem and metrics relevant to these problems, namely accuracy, confusion matrix, and Receiver Operating Characteristic (ROC) curve have been utilized for showing efficacy of the solutions. In fraud detection, recall rate is more important than the overall accuracy and precision. Accuracy alone cannot reflect the quality of the algorithms because by simply predicting that all transactions are good events and not detecting even a single fraud can still get high accuracy; for example, if the ratio of fraud against normal is around 1.2% in the data. The accuracy can be 98.8% if simply guessing every transaction is normal. Metric like balanced error rate (BER), which is average of FPR and FNR, is the mean of the errors on each class and would be more appropriate for skewed domain like this. Matthews correlation coefficient (MCC) is also a balanced metric for classification performance [43].

In conventional metrics, each misclassification has same cost. However, in domain of FDS, frauds of small and big amounts must not be treated with equal importance. Therefore, the cost of a fraud is often assumed to be equal to the transaction amount [53].

Along with, cost should also include the time taken by the detection system to react. The shorter is the reaction time, the larger is the number of frauds that it is possible to prevent.

In [6], a new metric "misclassification cost" (false-positive and false-negative error costs) has been defined. It has been used to modify AdaBoost learning algorithm's internal heuristics to cost instead of accuracy. Authors in [17] redefined "False Negatives (FN)" metric as amount of transaction that was misclassified. It has also been concluded that a false-negative error is usually costlier than a false-

positive error in case of fraud detection. Another challenge in correctly defining metrics for fraud detection is that costs change from case to case and over time.

Therefore, there is still no standard evaluation criterion for assessing and comparing the results of fraud detection systems. Techniques proposed in literature focused on solving any one issue related with this problem; for instance, it may be either concept drift or imbalanced dataset. An ideal FDS need to address all the challenges discussed in Sect. 1.2. It should be able to provide empirical answers to questions like:

- What should be the training set size for a perfect computational model of FDS?
- What is the correct pre-processing method, if any, to be applied on raw data from any source?
- Which metric or set of metrics can best evaluate the FDS across all cases?
- What should be the frequency of retraining the computational model?
- Should give guaranteed upper bound on false alarm rates.
- Minimize false negatives.

5.6 Next-Generation Computational Model for Credit Card Fraud Detection

Much work has been done in developing techniques for the detection of frauds; however, there is still more to do. Learning from nonstationary data stream with skewed class distribution with real-time requirements along with low false positive and high true negatives ratio is a relatively recent domain. After critical examination of the body of work done on design of computational models for FDS, following are few directions in which further progress is required:

1. Support interactive dashboards to quickly spot anomalous transactions.
2. Support for traceback and postfraud evidence gathering.
3. Be agile to discover and resist emerging fraud strategies.
4. Adapt techniques from Big Data and streaming Analytics to combat fraud detection challenges.
5. Formal feature engineering models for building effective classifiers need to be designed.
6. Domain-specific "end-to-end" performance measures like time to detect and recovery percentages need to be related to standard detection metrics.

6 Conclusions

Credit card frauds are a problem of recent concern due to a rapid rise in credit card-based transactions. Many machine learning-based computational models have been proposed to design an effective credit card fraud detection system. In this chapter, most popular models proposed in last two and half decades have been analyzed. It

was found that existing fraud detection systems suffer from problems like limited knowledge about credit card-based payment processing, nonexistence of standard algorithm, suitable metrics, and high rate of false-positive alarms. Over and above, there are no credit card benchmark datasets that can be tested for effectiveness of newer models. Technologies from streaming data and big data analytics have not yet been applied to this domain and can be explored.

References

1. Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011). Data mining for credit card fraud: A comparative study. *Decision Support Systems, 50*(3), 602–613.
2. Gupta, B. B., & Quamara, M. A. (2018). Taxonomy of various attacks on smart card–based applications and countermeasures. *Concurrency Computation Practice Experience*, e4993.
3. Ghosh, R. (1994). Credit card fraud detection with a neural-network. In *Proceedings of the Twenty-Seventh Hawaii International Conference on System Sciences*, Wailea, HI (pp. 621–630).
4. Richardson, R. (1997). Neural networks compared to statistical techniques. In *Proceedings of the IEEE/IAFE 1997 Computational Intelligence for Financial Engineering (CIFEr)*, New York City, NY (pp. 89–95).
5. Hanagandi, V., Dhar, A., & Buescher, K. (1996). Density-based clustering and radial basis function modeling to generate credit card fraud scores. In *IEEE/IAFE Conference on Computational Intelligence for Financial Engineering (CIFEr)*, New York City, NY (pp. 247–251).
6. Chan, P. K., Fan, W., Prodromidis, A. L., & Stolfo, S. J. (1999). Distributed data mining in credit card fraud detection. *IEEE Intelligent Systems and Their Applications, 14*(6), 67–74.
7. Hand, D. J., & Blunt, G. (2001). Prospecting for gems in credit card data. *IMA Journal of Management Mathematics, 12*(2), 173–200.
8. Bolton, R. J., & Hand, D. J. (2001) Unsupervised profiling methods for fraud detection. In *Conference on Credit Scoring and Credit Control*.
9. Syeda, M., Zhang, Y.-Q., & Pan, Y. (2002) Parallel granular neural networks for fast credit card fraud detection. In *IEEE World Congress on Computational Intelligence Proceedings* (Cat. No.02CH37291), Honolulu, HI (Vol. 1, pp. 572–577).
10. Bolton, R. J., & Hand, D. J. (2002). Statistical fraud detection: A review. *Statistical Science, 17*. https://doi.org/10.1214/ss/1042727940.
11. McCarty, B. (2003). Automated identity theft. *IEEE Security & Privacy, 99*(5), 89–92.
12. Kou, Y., Lu, C.-T., Sirwongwattana, S., & Huang, Y.-P. (2004) Survey of fraud detection techniques. In: *IEEE International Conference on Networking, Sensing and Control*, Taipei (Vol. 2, pp. 749–754).
13. Leung, A., Yan, Z., & Fong, S. (2004). On designing a flexible e-payment system with fraud detection capability. In *Proceedings. IEEE International Conference on e-Commerce Technology*, San Diego, CA (pp. 236–243).
14. Panigrahi, S., Kundu, A., Sural, S., & Majumdar, A. K. (2009). Credit card fraud detection: A fusion approach using Dempster-Shafer theory and Bayesian learning. *Information Fusion, 10*, 354–363.
15. Quah, J. T. S., & Sriganesh, M. (2007). Real time credit card fraud detection using computational intelligence. In *International Joint Conference on Neural Networks*, Orlando, FL (pp. 863–868).
16. Brabazon, A., Cahill, J., Keenan, P., & Walsh, D. (2010). Identifying online credit card fraud using artificial immune systems. In *IEEE Congress on Evolutionary Computation*, Barcelona (pp. 1–7).

17. Bahnsen, A. C., Stojanovic, A., Aouada, D., & Ottersten, B. (2013) Cost sensitive credit card fraud detection using Bayes minimum risk. In *12th International Conference on Machine Learning and Applications*, Miami, FL (pp. 333–338).
18. Duman, E., & Özçelik, M. (2011). Detecting credit card fraud by genetic algorithm and scatter search. *Expert Systems with Applications, 38*, 13057–13063. https://doi.org/10.1016/j.eswa.2011.04.110.
19. Zareapoor, M., Seeja, K. R., & Alam, A. (2012). Analysis on credit card fraud detection techniques: Based on certain design criteria. *International Journal of Computer Applications, 52*, 35–42. https://doi.org/10.5120/8184-1538.
20. Wong, N., Ray, P., Stephens, G., & Lewis, L. (2012). Artificial immune systems for the detection of credit card fraud: An architecture, prototype and preliminary results. *International Journal of Information Systems, 22*, 53–76.
21. Lei, O. Z., & Ghorbani, A. A. (2012). Improved competitive learning neural networks for network intrusion and fraud detection. *Neurocomputing, 75*(1), 135–145.
22. Zanin, M., Romance, M., Moral, S., & Criado, R. (2018). Credit card fraud detection through parenclitic network analysis. *Complexity*. Article ID 5764370, 9 pp.
23. Rajeshwari, U., & Babu, B. S. (2016). Real-time credit card fraud detection using streaming analytics. In *2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, Bangalore (pp. 439–444).
24. Zeager, M. F., Sridhar, A., Fogal, N., Adams, S., Brown, D. E., & Beling, P. A. (2017). Adversarial learning in credit card fraud detection. In *Systems and Information Engineering Design Symposium (SIEDS)*, Charlottesville, VA (pp. 112–116).
25. Awoyemi, J. O., Adetunmbi, A. O., & Oluwadare, S. A. (2017). Credit card fraud detection using machine learning techniques: A comparative analysis. In *International Conference on Computing Networking and Informatics (ICCNI)*, Lagos (pp. 1–9).
26. Carneiro, N., Figueira, G., & Costa, M. (2017). A data mining-based system for credit-card fraud detection in e-tail. *Decision Support Systems*. https://doi.org/10.1016/j.dss.2017.01.002.
27. Zheng, L., Liu, G., Luan, W., Li, Z., Zhang, Y., Yan, C., et al. (2018). A new credit card fraud detecting method based on behavior certificate. In *IEEE 15th International Conference on Networking, Sensing and Control (ICNSC)*, Zhuhai (pp. 1–6).
28. Cody, T., Adams, S., & Beling, P. A. (2018). A utilitarian approach to adversarial learning in credit card fraud detection. In *Systems and Information Engineering Design Symposium (SIEDS)*, Charlottesville, VA (pp. 237–242).
29. Patil, S., Nemade, V., & Soni, P. K. (2018). Predictive modelling for credit card fraud detection using data analytics. In *International Conference on Computational Intelligence and Data Science*.
30. Roy, A., Sun, J., Mahoney, R., Alonzi, L., Adams, S., & Beling, P. (2018). Deep learning detecting fraud in credit card transactions. In *Systems and Information Engineering Design Symposium (SIEDS)*, Charlottesville, VA (pp. 129–134).
31. de Sá, A. G. C., Pereira, A. C. M., & Pappa, G. L. (2018). A customized classification algorithm for credit card fraud detection. *Engineering Applications of Artificial Intelligence, 72*, 21–29.
32. Dhankhad, S., Mohammed, E., & Far, B. (2018). Supervised machine learning algorithms for credit card fraudulent transaction detection: A comparative study. In *IEEE International Conference on Information Reuse and Integration (IRI)*, Salt Lake City, UT (pp. 122–125).
33. Wang, Y., Adams, S. C., Beling, P. A., Greenspan, S., Rajagopalan, S., Velez-Rojas, M. C., et al. (2018). Privacy preserving distributed deep learning and its application in credit card fraud detection. In 17th *IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering* (TrustCom/BigDataSE), New York, NY (pp. 1070–1078).
34. Barker, K. J., D'amato, J., & Sheridon, P. (2008). Credit card fraud: Awareness and prevention. *Journal of Financial Crime, 15*(4), 398–410.
35. Abdallah, A., Maarof, M. A., & Zainal, A. (2016). Fraud detection system. *Journal of Network and Computer Applications, 68*, 90–113.
36. Dheepa, V., & Dhanapal, R. (2009). Analysis of credit card fraud detection methods. *International Journal of Recent Trends in Engineering, 2*(3), 126.

37. Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., & Bontempi, G. (2018). Credit card fraud detection: A realistic modeling and a novel learning strategy. *IEEE Transactions on Neural Networks and Learning Systems, 29*(8), 3784–3797.
38. Subbulakshmi, T., Mathew, G., & Shalinie, S. M. (2010). Real time classification and clustering of ids alerts using machine learning algorithms. *International Journal of Artificial & Application, 1*(1), 20.
39. Sorournejad, S., Zojaji, Z., Atani, R. E., & Monadjemi, A. H. (2016). A survey of credit card fraud detection techniques: Data and technique oriented perspective. *ArXiv preprint*, arXiv:1611.06439.
40. Dal Pozzolo, A., Caelen, O., Le Borgne, Y.-A., Waterschoot, S., & Bontempi, G. (2014). Learned lessons in credit card fraud detection from a practitioner perspective. *Expert Systems with Applications, 41*, 4915–4928.
41. Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., & Bontempi, G. (2015). Credit card fraud detection and concept-drift adaptation with delayed supervised information. In *International Joint Conference on Neural Networks (IJCNN)*, Killarney (pp. 1–8).
42. Dal Pozzolo, A. (2015). *Adaptive machine learning for credit card fraud detection* [online]. Retrieved from http://difusion.ulb.ac.be/vufind/Record/ULB
43. Randhawa, K., Loo, C. K., Seera, M., Lim, C. P., & Nandi, A. K. (2018). Credit card fraud detection using AdaBoost and majority voting. *IEEE Access, 6*, 14277–14284.
44. Zheng, L., Liu, G., Yan, C., & Jiang, C. (2018). Transaction fraud detection based on total order relation and behavior diversity. *IEEE Transactions on Computational Social Systems, 5*(3), 796–806.
45. Newman, D. J., & Asuncion, A. (2007) *UCI machine learning repository*. Transformed datasets are available at http://www.ulb.ac.be/di/map/adalpozz/imbalanced-datasets.zip.
46. Zaslavsky, V., & Strizhak, A. (2006). Credit card fraud detection using self-organizing maps. *International Journal Information & Security, 18*, 48–63.
47. Hssina, B., Merbouha, A., Ezzikouri, H., & Erritali, M. (2014) A comparative study of decision tree ID3 and C4.5. *International Journal of Advanced Computer Science and Applications* (Special Issue on Advances in Vehicular Ad Hoc Networking and Applications).
48. De Castro, L. N., & Timmis, J. (2002). *Artificial immune systems: A new computational intelligence approach*. London: Springer.
49. Behera, T. K., & Panigrahi, S. (2015) Credit card fraud detection: A hybrid approach using fuzzy clustering and neural network. In *Second International Conference on Advances in Computing and Communication Engineering*, Dehradun (pp. 494–499).
50. Heckerman, D. (1995). *A tutorial on learning with Bayesian*. Technical report, MSRTR-95-06. Redmond, WA: Microsoft Research.
51. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Philip Kegelmeyer, W. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*, 321–357.
52. Padmaja, T. M., Dhulipalla, N., Krishna, P. R., Bapi, R. S., & Laha, A. (2007). *An unbalanced data classification model using hybrid sampling technique for fraud detection* (Lecture Notes in Computer Science) (Vol. 4815). Berlin: Springer.
53. Elkan, C. (2001). The foundations of cost-sensitive learning. In *International Joint Conference on Artificial Intelligence*, Citeseer (Vol. 17, pp. 973–978).