LITERATURE SURVEY

CHAPTER 2

LITERATURE SURVEY

2.1 Introduction

Fraud detection is a critical part of the measures implemented for maintaining an attack tolerant database system. Though database management systems can provide intrusion prevention up to a certain extent by virtue of traditional access control mechanisms, they would not be sufficient for protection against syntactically correct but semantically damaging transactions. Chung et al. [23] bring out that misuse detection in database systems has not been adequately addressed and propose DEMIDS, which can derive user profiles from database audit logs. The field of Game theory has been explored for problems ranging from auctions to chess and its application to the domain of Information Warfare seems promising. Samuel et al bring out the role of game theory in information warfare [24]. They highlight that one can utilize well-developed Game theory algorithms to predict future attacks and the differences and challenges in this domain as compared to traditional games like chess, such as limited examples, multiple simultaneous moves and no time constraints [24].

2.2 Credit Card Fraud Detection using Various Methods and Techniques

Credit card fraud detection has drawn lot of interest and a number of techniques, with special emphasis on data mining and neural networks, have been

proposed to counter fraud in this field. Vatsa et al. [2] determine the effectiveness of neural network for credit card fraud detection. The neural network used for this study is the P-RCE (Restricted Coulomb Energy) neural network. The authors concluded that it was possible to achieve a reduction of 20% to 40% in the total fraud losses. Aleskerov et al. [25] presented CARDWATCH, a database mining system based on a neural network learning module. The system trains a neural network with the past data of a particular customer, which can then be used to process the current spending behavior and detect anomalies and they assume that since the normal behavior of the thief is to purchase as much as possible in limited time, the anomaly in transactions will most probably be detected.

In case deviations appear, the suspicious transactions can be taken under special consideration. However, it is not realistic to assume that with this procedure every fraudulent can be detected, because a customer may want to buy an unusual product or the card number thief may fit into the customers profile. But since the normal behavior of a thief is to purchase as much as possible in a limited time, the anomaly in the transactions will most probably be detected.

The learning rule of a network maybe selected in the current CARDWATCH version among conjugate gradient, back propagation with momentum and batch back propagation with momentum. The back propagation learning rule is standard learning technique. It performs a gradient descent in the error/weight space. To improve the efficiency, a momentum term is introduced which moves the correction of the weights in the direction complaint with the last weight correction. Batch back propagation with momentum is a modification of back propagation with momentum where the weights are changed once per epoch i.e.) after all the data patterns have been

processed. The Conjugate gradient method is an advanced optimization procedure which guarantees to locate the minimum of a quadratic function; for non-quadratic functions a criterion for convergence is required.

Chan et al divide a large data set of transactions into smaller subsets and then apply the mining techniques in parallel in a distributed data mining approach [26]. The resultant base models are then combined to generate a meta-classifier. More recently, Syeda et al have discussed the use of parallel granular neural networks for fast credit card fraud detection [27]. The Parallel Granular Neural Network (PGNN) aims at speeding up the data mining and knowledge discovery process. Suvasini panoramic et al. [21] suggested a fusion approach to find the suspicion level of the transaction. In this process, a hybrid approach is derived that combines both the supervised and unsupervised learning approaches.

This system consist of four components, namely, rule based filter, Dempster-shafer adder, transaction history database and Bayesian learner. In the rule based component, they determine the suspicion level of each incoming transaction based on the extent of its deviation from good pattern. Dempster –Shafer's theory is used to combine multiple such evidences and an initial belief is computed. The transactions are classified as normal, abnormal or suspicious depending on this initial belief. Once a transaction is found to be suspicious, belief is further strengthened or weakened according to its similarity with fraudulent or genuine transactions history using Bayesian learning. Extensive simulation with stochastic models shows that fusion of different evidences has a very high positive impact on the performance of a credit card fraud detection system as compared to other methods.

In [2] the fraud detection system comprises of two layers, the Rule-based component and the Game-theoretic component. The First Layer of this architecture uses generic as well as customer-specific rules to calculate the overall suspicion score for a transaction that is submitted. Here, it assigns weights to the different attributes of a transaction on the same card number. Transactions scoring high due to attributes such as ‘high value’, ‘sale-able item’, ‘address mismatch’, etc, may trigger an alarm albeit the possibility of it being false cannot be ruled out. The main idea is that given a transaction and a specific user, what confidence measure can be assigned for the transaction to be from the genuine cardholder. Hence, the First Layer flags a transaction as ’suspect’ if it crosses a user-defined threshold level. This introduces a trade-off between false positives (when the threshold is low) and more seriously, false negatives. The second tier is the Game-theoretic component of the model.

2.3 Credit Card Fraud Detection in Unbalanced Datasets

The development process of the fraud detection system experiences some technical problems caused by peculiar characteristics of data, such as large volume of data, skewed distribution, irregular cost of transaction, and highly overlapped classes [26]. Most serious one is the construction of training data set for the classifier. Especially for the classifier based on an unstable learning algorithm such as neural networks, decision tree or decision rule, its learning result undergoes significant changes in response to small changes in the training data set [28]. For these reasons, generation of the unbiased data set for training the classifier is essential in the first stage of modeling fraudulent behaviors. But, it is also not so easy because of extremely skewed distribution of data as well as their huge size.

The basic objective of [29] is to improve the performance of the neural classifier which is constructed with training data set that contains only a small part of real transaction data and is very extremely biased to fraudulent transactions. As the bias of training data set is inevitable in constructing the neural classifier for the detection of fraudulent transactions, the process is designed with fraud density map using analyzed information over the real data. From the intuitions that the input transaction can be regarded suspicious if fraud-ridden area in the input space and the fraud density represents the probability of fraud occurrence, it takes the fraud ratio of neighborhood around the point represented by a input vector in the input data space as a confidence value, which the fraud score of the neural classifier adjusted.

Maruthi et al. [30] considers fraud detection as an unbalanced data classification problem where the majority samples (non-fraud samples) outnumber the minority samples (fraud samples). Usually, the classification algorithms exhibit poor performance while dealing with unbalanced datasets and results are biased towards the majority class. Hence, an appropriate model is needed to classify unbalanced data, especially for the fraud detection problem. For these types of problems, the accuracy of the classifier is not trusted because the cost associated with fraud sample being predicted as a non-fraud sample is very high. [30] Proposed a model for fraud detection that uses hybrid sampling technique, which is a combination of random under-sampling and Synthetic Minority Over-sampling Technique (SMOTE) [31] to oversample the data. It uses Value Difference Metric (VDM) as a distance measure in SMOTE. The figure 2.1 presents a hybrid sampling model for fraud detection.

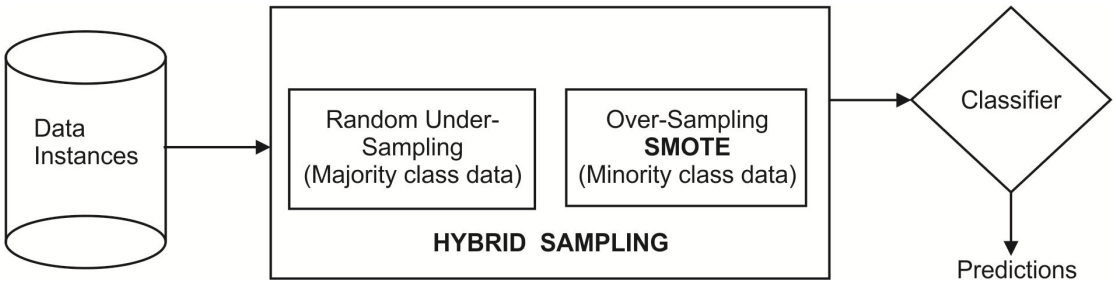

Figure 2.1 Hybrid Sampling Model for Fraud Detection

Some of the techniques available to handle the unbalanced data are: (a) oneclass classification, (b) sampling techniques (c) ensembling of classifiers and (d) rule based approaches. Foster [32] presented a review on the issues related to unbalanced data classification. From the experiments on 25 unbalanced data sets at different unbalanced levels, it is concluded that the natural distribution is not usually the best distribution for learning. Kubat and Matwin [33] did selective under-sampling of majority class by keeping minority classes fixed. They categorized the minority samples into some noise overlapping, the positive class decision region, borderline samples, redundant samples and safe samples. By using tomek links concept, which is a type of data cleaning procedure used for under-sampling, they deleted the borderline majority samples. Kubat [34] proposed SHRINK system which searches for best positive regions among the overlapping regions of the majority and minority classes.

Chawla et al [31] proposed Synthetic Minority Over-Sampling Technique (SMOTE). It is an over-sampling approach in which the minority sample is over-sampled by creating synthetic samples rather than by oversampling with replacement. The minority class is over-sampled by taking each minority class sample and introducing synthetic samples along the line segments joining any/all of the k minority class' nearest neighbors. Depending upon the amount of over-sampling

required, neighbors from the k nearest neighbors are randomly chosen. This approach effectively forces the decision region of the minority class to become more general. Study of “whether over-sampling is more effective than under-sampling” and “which over-sampling or under-sampling rate should be used” was done by Estabrooks et al [35], who concluded that combining different expressions of the resampling approach is an effective solution.

2.4 Meta Classifier System for Credit Card Fraud Detection

An excellent survey of work on fraud detection has recently been done by Clifton [3]. This work defines the professional fraudster, formalizes the main types and subtypes of known fraud and presents the nature of data evidence collected within affected industries. Stolfo et al [36] outlined a meta classifier system for detecting credit card fraud by merging the results obtained from the local fraud detection tools at different corporate sites to yield a more accurate global tool. Similar kind of work has been carried out by Stolfo et al [37] and elaborated by Chan et al [26] .Their work described a more realistic cost model to accompany the different classification outcomes. Wheeler and Aitken [38] have also explored the combination of multiple classification rules. Clifton et al [39] proposed a fraud detection method, which uses stacking bagging approach to improve cost savings.

There are a number of techniques used by attackers to sniff the fixed credit card number used for authentication, and to use it for fraudulent payments. This is of course a direct consequence of the intrinsic weakness of the traditional credit card processing system, where the key used for authentication is long-term, semi-secret, transmitted over insecure channels, sometimes completely disclosed.

Consider that credit cards are still widely used in e-commerce [40] since it often happens that alternative secure payment methods are not applicable or preferred. A recent approach to contrasting the above problem is based on the concept of disposable-number credit cards. According to this scheme, issuer and customer agree on a number to use for the transaction, and then they discard it, generate a new number for the next transaction and so on. This way, sniffing an authentication number during a transaction does not give the attacker any useful credential.

There are commercial solutions based on the above scheme. Unfortunately, such solutions are either still insecure, when the cardholder gets the disposable number from the issuer Web site and authenticates itself by sensible data (like a standard credit card number) [41] or too expensive (and little friendly), when the generation of the new number is executed on board of a smartcard, and the issuer provides the cardholder with an additional device capable of displaying the disposable number. In principle, the extra cost related to the additional device could be eliminated by exploiting simple software solutions to display disposable numbers, interfaced with standard smart-card drivers. Anyway, the cost of (secure) smart cards is not irrelevant, so that a real applicability of the above strategy could be strongly related with the possibility of decreasing significantly also the cost of the card. Another related problem is that the number generation scheme should be enough secure, because, differently from schemes based on an extra communication channel (typically the Web), where numbers can be generated randomly by the issuer, necessarily there will be a mathematical link between a given disposable number and the successor one.

This mathematical link could be thus exploited by the attacker in order to predict new numbers on the basis of the past ones. As a consequence adequate efforts are necessary in order to design number generation schemes sufficiently robust, w.r.t. possible attacks. There are a number of research solutions [42, 44] addressing the above problem. The first approach of this type is presented in [43] that generate a new authentication number by encrypting in a smart card a set of possible restrictions describing some elements of the transaction itself. Starting from the consideration that encryption is too expensive to be realistically used in this context, the authors of [42] propose the use of context free grammars (thus not relying on any cryptographic algorithm) to generate disposable credit card numbers. Context free grammars present the property that the generation and validation of strings belonging to a given language can be done in polynomial time, but it is unfeasible to find the grammar given only the strings generated by it, since any conjectured grammar may fail on a new input string. However, the authors do not give any suggestion about how context free grammars have to be generated. As a consequence, an unlucky generation of the grammar may allow an attacker to easily guess the grammar. Moreover, there exists no theoretical result about how difficult is it to guess another string which belongs to the same language, thus showing the impossibility to guarantee the security of this technique. Also the authors of [44] propose a more efficient solution using cryptographic hash functions rather than encryption.

Wael et al [45] proposes a fraud engine based on an Artificial Neural Network (ANN). It was implemented on a Smart card in order to assess the performance and the general feasibility of this approach. The motivation was that, the intelligent behavior-based security mechanisms can provide added protection for critical

systems. Of particular interest is the real-time detection and reaction to fraudulent behaviors [46]. Any suspicious or unusual activities are captured and prevented instantly. With real artificial intelligence implemented using Neural Networks, behavior based security mechanisms promise to be at the same step as the attacker and not a step behind. Using this kind of approach to security brings it down to the personal level, meaning fraud should be detectable for every single user or customer depending on his usage characteristics.

There are already systems that are designed to analysis behavior to detect potential fraud. Such systems, known as fraud engines, are based on studying a certain behavior and reporting if a different behavior is detected. Neural Fraud Management Systems (NFMS) that are completely automated and state of-the-art integrated system of neural networks, fraud detection engines and automatic modeling systems.

Machine learning, anomaly-detection ANN based system can be used to address the shortcomings of rule-based systems. Rather than having to wait for a new attack to be detected and for a new rule to be written by an expert, these systems automatically and immediately detect unusual behavior for each user and for groups of users. Behavioral systems are inherently future proof as they can spot new types of attacks the first time that they are executed. An effective anomaly detection system relies on clustering algorithms that are based on artificial neural networks. A clustering algorithm groups similar transactions into a small number of clusters. Each cluster represents a common pattern of activity. Each time a new transaction is processed by the anomaly detection system, the system tries to fit it into an existing cluster. If a transaction does not fit into any cluster, it is classified as an anomaly.

D.Nali et al.[47] proposes a universal infrastructure and protocol for IDF detection, which is called CROO (Capture Resilient Online One-time password scheme). According to this proposal, each user must carry a personal device used to generate One-Time Passwords (OTPs) verified by online trusted parties. These OTP generation and verification procedures are universal, in the sense that they can be associated with any user transaction, regardless of the transaction's purpose (e.g. user identification, user authentication, or financial payment), associated credentials (e.g. driver's license or credit card), and online or on-site (e.g. point-of-sale) nature. For increased scalability, multiple OTP verification parties may be used. OTPs are not sent in clear text; they are used as keys to compute MACs of hashed unique transaction information (e.g. list of bought items). This allows OTP-verifying parties to confirm that given user credentials (i.e. OTP-based MACs) correspond to claimed hashed transaction details. Hashing transaction information increases user privacy. Online OTP-verifying parties detect IDF when OTPs of received user credentials or the associated transaction information do not have expected values. Each OTP is generated from a high-entropy non-verifiable text encrypted using a key derived from a user-chosen PIN; hence, possession of a user's personal device (or clone thereof) does not suffice to confirm guesses of the associated PIN, to recover the associated non-verifiable key, and generate correct OTPs. Since OTPs can only be verified by online parties, the proposed scheme turns off-line PIN guessing attacks against stolen or cloned personal devices into online OTP-guessing attacks that can be easily detected by online parties.

CROO provides means to both prevent IDF (by detecting IDF attempts), and limit its consequences when sophisticated IDF attacks have bypassed the aforementioned preventive measures.

2.5 Role of Support Vector Machine

Vapnik [48] describe a training algorithm for optimal margin classifiers in which they showed that maximizing the margin between training examples and class boundary amounts to minimizing the maximum loss with regards to the generalization performance of the classifier. This idea was initially explored because binary class optimal margin classifiers achieve errorless separation of the training data, given that separation is possible, and outliers are easily identified in such a classifier.

The first investigations into this type of algorithm were based on separable data sets, but, in 1995, Cortes and Vapnik [49] extended the algorithm to account for linearly inseparable data; attempts soon followed to also extend the results to multi-class classification problems. This type of learning machine was later dubbed the Support Vector Machine (SVM). The SVM is a machine learning technique with a strong and sound theoretical basis. It is interesting to note that, in most cases, researchers claim that SVMs match or outperform neural networks in classification problems. Methods like Time-Delay Neural Networks, Plate's method and NARX networks are examples of learning algorithms that are applicable to short time gap problems only. These will, in all probability, not perform very well in this specific problem. Many attempts have been made to address the long time lag problem. Gradient based methods, included simulated annealing, multi-grid random search, time-weighted pseudo-Newton optimization, and discrete error propagation. Simulated annealing performed the best on all their experimental problems, but it requires a lot more training time than the others, with training time also increasing as sequence length increases. As reported in earlier, most of these attempts to bridge long time lags can be outperformed by simple random weight guessing. Other

methods, for example unit addition and Kalman Filter RNNs, promises good results on longer time lags, but these are not applicable to problems with unspecified lag durations.

Information technologies, if properly used, can help detect and manage frauds in an efficient and effective manner [50]. Numerous methods and technologies have been proposed for the implementation of Fraud Management Systems (FMSs) [51]; among which rule-based scoring models, statistical tools, and Artificial Neural Networks (ANNs) are the common ones.

Although ANNs have some advanced features and are widely used for fraud detection, traditional back-propagation-based algorithms used for ANN training have the local optima problem which reduces the effectiveness of ANNs in detecting frauds. To alleviate the problem, this paper proposes an approach to training an ANN using an ant colony optimization technique, which can solve the local optima problem and improve the effectiveness of ANNs in fraud detection.

A scoring model is a set of rules that describes the characteristics of some particular known fraud types. Scoring models work well when the fraud types are simple and manageable in the scale and complexity of rule parameters. Even for simple fraud types, however, they can be very laborious to define all the rules. In addition, scoring models have no self-learning capability and are unable to swiftly adapt to new fraud types.

Several statistical tools are used to detect frauds. Least-squares regression analysis is used in fraud audits. In this regard, practical guidelines are provided to those auditors who are interested in detecting frauds without focusing on the

associated mathematical details. In [52], a hybrid financial analysis model is presented, involving static and trend analysis models to construct and train a back-propagation neural network model. The experiments in [52] show that the proposed model not only provides a high predication rate but also outperforms other models including discriminant analyses, decision trees, and back-propagation neural networks alone.

However, pure statistical tools can hardly provide an adaptive learning capability in fraud management. Artificial neural networks (ANNs) with self-learning capabilities are proposed to overcome the above problems. ANNs learn by experience, generalize from previous experiences to new ones, and can make decisions. ANNs are motivated by information-processing units as neurons in the human brain that a neural network is made up of artificial neurons [52]. A neural network can be thought of as a black box non-parametric classifier. Neural networks are therefore more flexible. ANNs have become an important technology for pattern recognition and implementation of a FMS.

Despite of their growing popularity, the performance of ANNs largely depends on how the networks are trained [53]. Most of the current ANN training for fraud detection is designed based on a Back-Propagation (BP) algorithm. This type of algorithms, however, has a well-known problem, that is, difficulty in escaping local optima. This is because all BP algorithms used for training ANNs initialize the starting point in an n-dimensional space randomly. If the starting point is located in a local valley, a BP algorithm probably converges on a local solution. Many modifications have been made to overcome this problem by finding the global solution [53].Ant Colony Optimization (ACO) is a metaheuristic technique based on

the research of ant group behaviors in the natural world [54]. It imitates ants' behaviors in establishing shortest paths from their nest to feeding sources and back. Individual ants are supposed to interact with each other by some chemical pheromone released by them.

When an ant finds a food source, it releases chemical pheromone on the ground. The quantity of pheromone depends on the quantity and quality of the food source. The pheromone vanishes over time. When another ant looks for food, it moves in a random manner basically. However, if it detects chemical pheromone in the environment around, it will have higher probability to move toward the direction with denser pheromone. This in turn reinforces the trail with more pheromone, and attracts more ants to the food source. The indirect communication between the ants via the pheromone trails allows them to find shortest paths between their nest and food sources.

The collective behavior brings ACO with characteristics of positive feedback, parallel computing, robustness, and global optimization [54]. Recently, an approach to fraud detection utilizes ant colony algorithm to optimize ANN settings. It actually includes three steps:

1. Initial authentication (often by means of PINs and passwords)
2. Automatic fraud detection by ANN
3. Further review by human users

Step 1 is a preliminary process to check the user identity and authentication. It can use PINs, passwords, or IP addresses in the Internet; fingerprint in some terminals; and SIM card identity or calling site in telecommunication. This step can reveal a large number of identity misuse problems (such as identity theft).

Step 2 is the core part of this approach. While step 1 is a signature-based analysis process, step 2 is a behavior-based process that needs more intelligent methods to analyze the behavior data. Users' transaction data (like telecom subscribers' calling detail records) are collected and analyzed in order to build up their profiles. If any abnormal transaction occurs, the case will be highlighted for manual review in step 3.

However, abnormal transactions are just potential fraudulent events, and sometimes they are just occasional usage of legal users. So in step 3, those abnormal transactions detected in step 2 are sent to the relevant employees for further review. Thus this three-step approach forms a system approach that provides overall controls to the person in charge of fraud detection.

Artificial Immune Systems (AIS) represent an important strategy inspired by biological systems. There is significant growth in the applications of immune system models in various domains and across many fields such as computer security, optimization, robotics, fault detection etc. But for fraud detection, there are only few attempts in this area.

Gadi et al. has demonstrated the AIS algorithm outperforms other traditional techniques for credit card fraud detection which the data are highly imbalanced [55].Another work done by Brabazon et al [56] also demonstrated artificial immune technique can be used for online credit card fraud prevention[57]. In their paper, they investigate the effectiveness of AIS for credit card fraud detection. The results suggest AIS algorithm have potential for inclusion in fraud detection system. Huang [52] proposes an immune inspired adaptive online fraud detection system to counter this

threat. This proposed system has two layers: the innate layer that implements the idea of Dendritic Cell Analogy (DCA), and the adaptive layer that implements the Dynamic Clonal Selection Algorithm (DCSA) and the Receptor Density Algorithm (RDA).

Rather than detecting credit card fraud by past transaction data, Chen et al. [57] proposed a novel approach to solve the fraud problem. They proposed to develop a personalized model to detect fraud. One can first gather personal transaction data of users by an online, self-completion questionnaire system.

The gathered Questionnaire-Responded Transaction (QRT) data from the online system are considered as the transaction records and are utilized to build up a personalized model, which is sequentially in use to predict whether a new, actual transaction is a fraud or not. Since the illegal user's consumer behavior is usually dissimilar to the cardholder, the fraud can be avoided from initial use of a credit card, even without any transaction data. The QRT approach is promising. However, there are still some problems needed investigating. One of the most important issues regarding the QRT approach is how to predict accurately with only few data, say 100 to 200, since the users are usually not willing to answer too many questions. In this paper, the influences of data number and data distribution on the prediction accuracy are examined. Several typical ways for improving the prediction accuracy are also employed to study their influences. Figure 2.2 represents the QRT approach to fraud detection.

Some useful tools have been successfully applied to fraud detection, such as, artificial neural network, learning machines, and so forth. Recently, support vector

machines (SVMs) [48] are up-and-coming as a powerful machine learning technique to classify and do regression. The SVM has already been successfully used for a wide variety of problems, like bio-informatics, natural language learning, text mining, pattern recognition, and so forth. The SVM has some desirable properties that make it a very powerful technique to use computational complexity is enhanced by the use of the kernel trick, over-fitting is avoided by classifying with a maximum margin hyper-plane, and only a small subset of the training set is needed to separate the different classes of the problem.

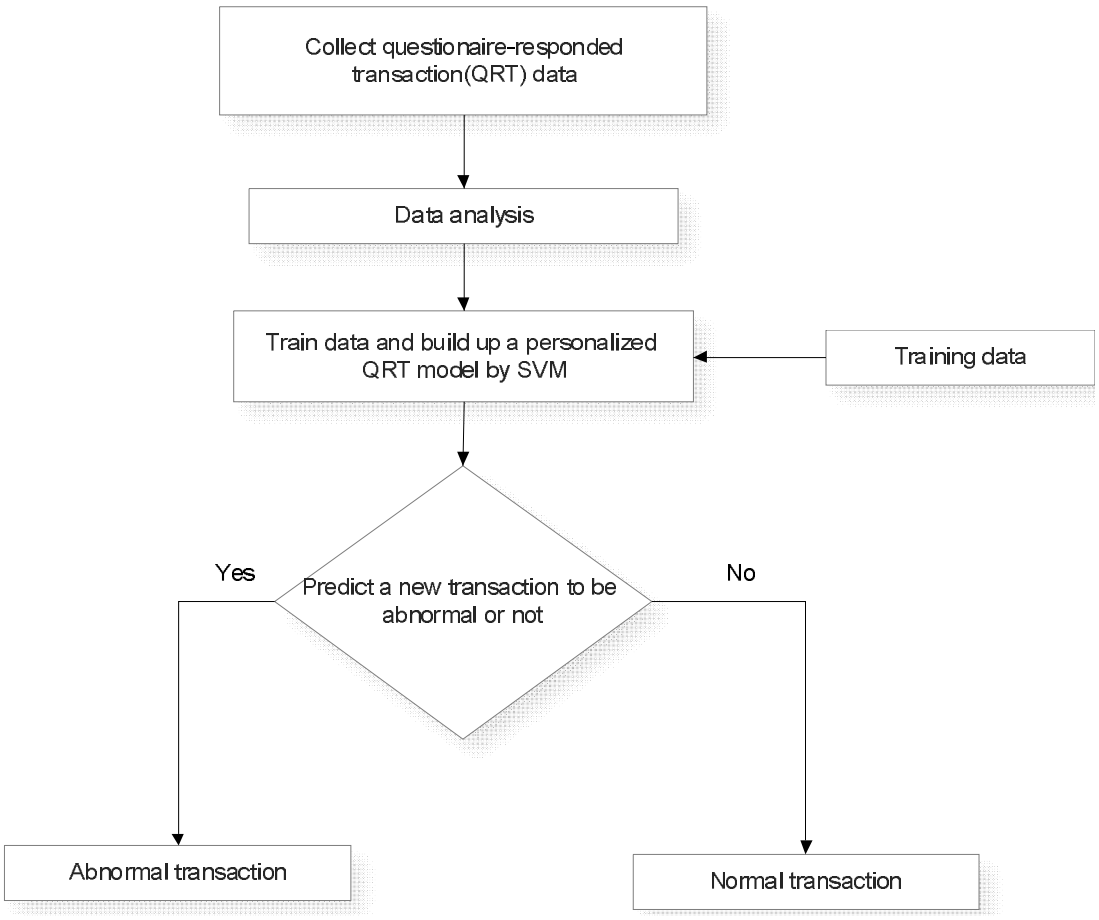

Figure 2.2 QRT Approach for Predicting Frauds

Currently available Fraud Detection Systems (FDS) are either misuse-based or anomaly-based. A misuse-based FDS cannot detect any new fraud pattern. On the other hand, anomaly based detection systems often raise a large number of false alarms. It uses sequence alignment techniques, normally used to solve problems in the bioinformatics domain, to determine possible cases of fraudulent transactions. This model is based on the strengths of both anomaly and misuse based detection models. This feature of the FDS helps to overcome the drawbacks of existing fraud detection systems and makes it an efficient fraud detection model. Sequence alignment is used to determine similarity of an incoming sequence of transactions to both a genuine card holder's sequence as well as to sequences generated by a validated fraud model. The scores from these two stages are combined to determine if a transaction is genuine or not.

Given that neither behavioral models nor transaction-level classification are foolproof strategies for detecting fraud, it is important to find other strategies which may be able to meliorate their weaknesses. All these strategies will ultimately be used in parallel, together with any existing rule-based detection system (e.g. for collision detection). An obvious extension to transaction-level classification is to aggregate information over a succession of transactions, or a period of time. This is indeed the approach often taken by designers of commercial fraud detection systems.

Previous published approaches to credit card transaction fraud, based on supervised classification, concentrate on classifying individual transactions rather than account level detection.

2.6 Role of Clustering and Outlier Detection

Ren [58] proposes an efficient outlier detection method with clusters as by-product, which works efficiently for large datasets. The contributions include: a) introducing a Local Connective Factor (LCF); b) Based on LCF, an outlier detection method which can efficiently detect outliers and group data into clusters in a one-time process is proposed. This method does not require the beforehand clustering process, which is the first step in other state-of-the-art clustering-based outlier detection methods; c) The performance of this method is further improved by means of a vertical data representation, Ptrees.

LCF indicates the degree at which a point locally connects with other points in a dataset. Unlike the current cluster-based outlier-detection approaches, this method detects outliers and groups data into clusters in a one-time process. This method has two advantages. First, without Clustering beforehand, the outlier-detection process is speed up significantly. Second, although the main purpose is to find outliers, the method can group data into clusters to some degree as well. A vertical data representation, P-Tree, is used to speed up this method further. The calculation of LCF using P-Trees is very fast. P-trees are very efficient for neighborhood-search problems; they use mostly logical operations to accomplish the task. P-trees can also be used as self-indexes for certain subsets of the data. In this paper, P-trees are used as indexes for the unprocessed subset, clustered subset and outlier subset. Pruning is efficiently executed on these index P-trees.

Recently, spectral clustering has wide application in pattern recognition and data mining because it can obtain global optima solution and adapt to sample spaces with any shape. It is simple to implement and can be solved efficiently by standard

linear algebra software. Through spectral method the information of feature space with eigenvectors is used rather than that of the whole dataset to obtain stable clusters. Manzoor et al.[59] introduces the cluster-based local outlier factor to identify and find the outliers in dataset. It proposes an outlier detection algorithm based on spectral clustering. Experimental results show that it outperforms the traditional K-means based algorithm.

Syeda et al [60] have used parallel granular neural network for improving the speed of data mining and knowledge discovery process for credit card fraud detection. But it could achieve reasonable speed up to 10 processors only, more number of processors introduces load imbalance problem. Chiu et al [61] have proposed web-services based collaborative scheme for fraud detection in the banking industry. The proposed scheme supports the sharing of knowledge about fraud pattern with the participant banks in a heterogeneous and distributed environment. Phua et al [3] have done an extensive survey of existing data mining based FDSs. Abhinav srivastava et al [62] have proposed a Hidden Markov model for credit card fraud detection.

In this Model, the sequence of operations in credit card processing is modeled using a Hidden Markov Model (HMM).An HMM is initially trained with the normal behavior of cardholder. If an incoming transaction is not accepted by the trained HMM with sufficiently high probability, it is considered to be fraudulent. At the same time, this model tries to ensure the genuine transactions are not rejected.

Fan et al [26] suggest the application of distributed data mining in credit card fraud detection. To improve the efficiency of highly distributed databases and detection system, this approach uses boosting algorithm AdaCost, which uses large number of classifiers and requires more computational resources during detection.

Stolfo et al [63] suggest a credit card fraud detection system using meta learning techniques to learn models of fraudulent credit card transactions. Brause et al [64] combine advanced data mining techniques and neural network algorithms to achieve high fraud detection along with low false alarm. Naive Bayesian approach also suggested for credit card fraud detection. Naïve Bayesian algorithm is very effective in many real world data sets and is extremely efficient in linear attributes. Bayesian networks were more accurate and faster to train but are slower when applied to new instances. Jianyun et al [65] have presented a framework for detecting fraudulent transactions in an online system. That paper describes an FP tree based method to dynamically create user profile for the purpose of fraud detection. But this technique doesn't consider unusual patterns i.e. short term behavioral changes of genuine card holders. Wen-Fang et al [66] have proposed a research on credit card fraud detection model based on outlier detection mining on distance sum, which shows that outlier mining, can detect credit card fraud better than anomaly detection based on clustering. In this model, outlier mining was used in credit card fraud detection. Definitions of Distance based outliers are referred and the outlier mining algorithm was created. This model detects outlier sets by computing distance and setting threshold of outliers.

Experiment Process

1. Input was standardized by the standard deviation method.

2. Distances between two credit card transaction data set are computed and the sum value is obtained. If the sum value is large then other objects then it is considered as an outlier.

3. Set the threshold of outliers .Then it is compared and outlier set obtained.

C4.5 can output accurate predictions but scalability and efficiency problem occurs when applied to large data sets.

The existing credit card fraud detection techniques which use labeled data to train the classifiers are unable to detect new kinds of frauds. The common disadvantage of all supervised learning is that they require human involvement to optimize parameters. Decision tree do not require any parameter setting from the user and can be constructed faster compared to other techniques [67].

An efficient fraud detection system with BOAT algorithm [68], which is adaptive to the behavior changes by combining classification and clustering techniques. This is a two stage fraud detection system which compares the incoming transaction against the transaction history to identify the anomaly using BOAT algorithm in the first stage. In second stage to reduce the false alarm rate suspected anomalies are checked with the fraud history database and make sure that the detected anomalies are due to fraudulent transaction or any short term change in spending profile. In this work BOAT supports incremental update of transactional database and it handles maximum fraud coverage with high speed and less cost. Proposed model is evaluated on both synthetically generated and real life data and shows very good accuracy in detecting fraud transaction.

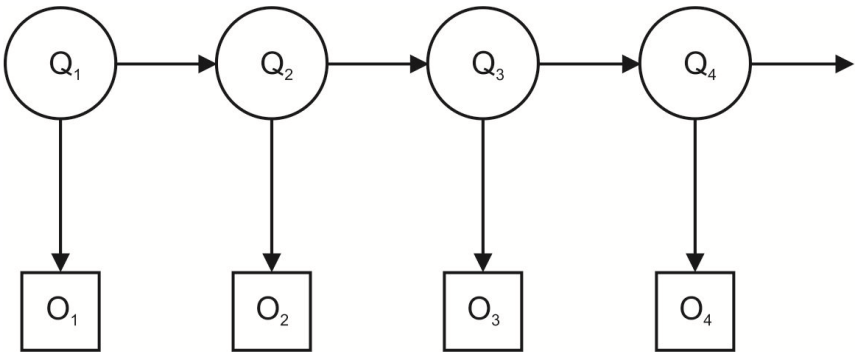

Figure 2.3 State of Hidden Markov Model

Hidden Markov model generate, observation symbols for online transaction. Observation probabilistic in an HMM based system is initially studies spending profile of the cardholder and checking an incoming transaction, against spending behavior of the cardholder. Figure 2.3 shows the states of a Hidden Markov model.

Fraud detection systems can be based on various approaches. The emphasis on fraud detection methodology is usually put upon supervised classification at transaction level that constructs an assignment procedure for new cases from the given training samples of fraudulent and non-fraudulent transactions.

In contrast to the supervised approach, fraud detection systems based on an unsupervised methodology monitor account activity and flag transactions inconsistent with an account's usual behavior observed over a period of time Some banks deploy the unsupervised methodology in the form of so-called ‘‘behavioral models" which build an individual profile for each account. This includes characteristics of account typical transaction activity, such as merchant types, time of day, monetary values, geographic locations, etc.

M. Krivko [69] presents the framework for a hybrid model for plastic card fraud detection systems. The proposed data-customized approach combines elements of supervised and unsupervised methodologies aiming to compensate for the individual deficiencies of the methods. applies fuzzy association rules in order to extract knowledge so that normal behavior patterns may be obtained in unlawful transactions from transactional credit card databases in order to detect and prevent fraud.

More recently, Syeda et al. [27] have suggested the use of parallel granular neural networks for speeding up the data mining and knowledge discovery process. Maes et al. [70] have outlined an automated credit card fraud detection system by ANN as well as Bayesian Belief Networks (BBN). They show that BBN gives better results related to fraud detection and the training period is faster whereas the actual detection process is substantially faster with ANN. The neural network based methods are, in general, fast but not so accurate. Re-training the neural networks is also a major bottleneck since the training time is quite high.

Chen et al. [57] proposed a novel method in which an online questionnaire is used to collect Questionnaire-Responded Transaction (QRT) data of users. A Support Vector Machine (SVM) is trained with this data and the QRT models are used to predict new transactions. Recently they presented a personalized approach for credit card fraud detection that employs both SVM and ANN. It tries to prevent fraud for users even without any transaction data. However, these systems are not fully automated and depend on the user's expertise level.

Some researchers have applied data mining for credit card fraud detection. Chan et al. [71] divides a large set of transactions into smaller subsets and then apply distributed data mining for building models of user behavior. The resultant base models are then combined to generate a meta-classifier for improving detection accuracy. Brause et al. [72] have explored the possibility of combining advanced data mining techniques and neural networks to obtain high fraud coverage along with a low false alarm rate.

Use of data mining is also elaborated in the work by Chiu and Tsai [73]. They consider web services for data exchange among banks. A Fraud Pattern Mining (FPM) algorithm has been developed for mining fraud association rules which give information regarding the features that exist in fraud transactions. Banks enhance their original fraud detection systems by using the new fraud patterns to prevent attacks. While data mining techniques are relatively accurate, they are inherently slow. Meta-learning is a general strategy that provides a means for combining and integrating a number of separately learned classifiers or models. A meta-learning system allows financial institutions to share their models of fraudulent transactions by exchanging classifier agents. Stolfo et al. [74] suggest a meta-learning technique to learn patterns of fraudulent credit card transactions. They apply four base classifiers, namely, ID3, CART, Bayes and RIPPER and use the class-combiner strategy to select the best classifier for meta-learning. It has been shown that meta-learning with Bayes gives good accuracy. Prodromidis and Stolfo [63] describe an artificial intelligence based approach that combines inductive learning algorithms and meta-learning methods to build accurate classification models for electronic fraud detection. The field of game theory has also been explored for credit card fraud detection.

Vatsa et al. [2] have modeled the interaction between an attacker and an FDS as a repeated game between two players, each trying to maximize its payoff. Such game-theoretic models make a number of assumptions, like availability of strategies, actions and payoffs to both the players, which are not often valid in practice. For example, it is quite unusual for a bank to advertise its strategies for fraud detection.

Some survey papers have been published which categorize, compare and summarize articles in the area of fraud detection. Phua et al. [3] did an extensive

survey of data mining based FDSs and presented a comprehensive report. Kou et al. [75] have reviewed the various fraud detection techniques including credit card fraud, telecommunication fraud as well as computer intrusion detection. Bolton and Hand [76] describe the tools available for statistical fraud detection and areas in which fraud detection technologies are most commonly used.

Majority of the FDSs as described above show a lot of variation in their accuracy. The main challenge identified by most of them is that the bulk of the transactions flagged as fraudulent by the FDSs are in fact genuine. A substantial amount of time and money is spent by bankers in investigating a large number of legitimate cases. It also causes customer inconvenience and potential dissatisfaction.

In credit card application, since occurrence of fraud is sparse, it involves detecting a relatively rare event from a very large collection of routine transactions. Axelsson [77] has pointed out that due to the base-rate fallacy problem; the factor limiting the performance of an intrusion detection system is not the ability to identify intrusive behavior correctly but its ability to minimize false alarms. While failure to detect a fraud causes direct loss to the company, follow up actions needed to pursue false alarms also tend to be costly. Any design choice that attempts to improve the rate of correct detection of fraud, usually causes a rise in the false alarms as well. One of the motivations of this research is to address this challenge. It is well known that every cardholder has a certain shopping behavior, which establishes an activity profile for him. Almost all the existing fraud detection techniques try to capture these behavioral patterns as rules and check for any violation in subsequent transactions. However, these rules are largely static in nature. As a result, they become ineffective when the cardholder develops new patterns of behavior that are not yet known to the

FDS. The goal of a reliable detection system is to learn the behavior of users dynamically so as to minimize its own loss. Thus, systems that cannot evolve or ‘‘learn”, may soon become outdated resulting in large number of false alarms. A fraudster can also attempt new types of attacks which should still get detected by the FDS. For example, a fraudster may aim at deriving maximum benefit either by making a few high value purchases or a large number of low value purchases in order to evade detection. Thus, there is a need for developing fraud detection systems which can integrate multiple evidences including patterns of genuine cardholders as well as that of fraudsters.

Suvasini Panigrahi et al.[21] proposes a novel approach for credit card fraud detection, which combines evidences from current as well as past behavior. The Fraud Detection System (FDS) consists of four components, namely, rule-based filter, Dempster–Shafer adder, transaction history database and Bayesian learner. In the rule-based component, determine the suspicion level of each incoming transaction based on the extent of its deviation from good pattern. Dempster–Shafer’s theory is used to combine multiple such evidences and an initial belief is computed. The transaction is classified as normal, abnormal or suspicious depending on this initial belief. Once a transaction is found to be suspicious, belief is further strengthened or weakened according to its similarity with fraudulent or genuine transaction history using Bayesian learning.

Kou et al.[75] highlight that an important advantage of data mining is that it can be used to develop a new class of models to identify new attacks before they can be detected by human experts. Phua et al. [8] point out that fraud detection has become one of the best established applications of data mining in both industry and

government. Various data mining techniques have been applied in FFD, such as neural networks, logistic regression models, the naïve Bayes method, and decision trees, among others.

Over the past few years, a number of review articles have appeared in conference or journal publications. Bolton and Hand [76], for example, have reviewed statistical methods of detecting fraud, including credit card fraud, money laundering, telecommunications fraud, etc. Zhang and Zhou [5]] have surveyed financial applications of data mining including stock market and bankruptcy predictions and fraud detection. Phua et al. [8] present a survey of data mining-based fraud detection research, including credit transaction fraud, telecoms subscription fraud, automobile insurance fraud and the like. Others have reviewed insurance fraud and financial statement fraud.

The method AdaCost was developed from Adaboost for credit card fraud detection, and resulted in the metaheuristic Cost Sensitive, which can be applied for many applications where there are different costs for false positive and false negative. Comparative studies between Neural Networks (NN) and Bayesian Networks (BN) in credit card fraud detection were reported, which favored the result of BN.

Statistical fraud detection methods have been divided into two broad categories: supervised and unsupervised. In supervised fraud detection methods, models are estimated based on the samples of fraudulent and legitimate transactions, to classify new transactions as fraudulent or legitimate. In unsupervised fraud detection, outliers or unusual transactions are identified as potential cases of

fraudulent transactions. Both these fraud detection methods predict the probability of fraud in any given transaction.

Predictive models for credit card fraud detection are in active use in practice. Considering the profusion of data mining techniques and applications in recent years, however, there have been relatively few reported studies of data mining for credit card fraud detection.

A recent paper [78] evaluates several techniques, including support vector machines and random forests for predicting credit card fraud. Their study focuses on the impact of aggregating transaction level data on fraud prediction performance. It examines aggregation over different time periods on two real-life datasets and finds that aggregation can be advantageous, with aggregation period length being an important factor. Aggregation was found to be especially effective with random forests. Random forests were noted to show better performance in relation to the other techniques, though logistic regression and support vector machines also performed well. Support vector machines and random forests are sophisticated data mining techniques which have been noted in recent years to show superior performance across different applications. The choice of these two techniques, together with logistic regression, for this study is based on their accessibility for practitioners, ease of use, and noted performance advantages in the literature.

SVMs are statistical learning techniques, with strong theoretical foundation and successful application in a range of problems [48]. They are closely related to neural networks, and through use of kernel functions, can be considered an alternate way to obtain neural network classifiers. Rather than minimizing empirical error on

training data, SVMs seek to minimize an upper bound on the generalization error. As compared with techniques like neural networks which are prone to local minima, overfitting and noise, SVMs can obtain global solutions with good generalization error. They are more convenient in application; with model selection built into the optimization procedure, and has also been found to outperform neural networks in classification problems**.** Appropriate parameter selection is, however, important to obtain good results with SVM.

Single decision tree models, though popular in data mining application for their simplicity and ease of use, can have instability and reliability issues. Ensemble methods provide a way to address such problems with individual classifiers and obtain good generalization performance. Various ensemble techniques have been developed, including mixture of experts, classifier combination, bagging, boosting, stacked generalization and stochastic gradient boosting. For decision trees, the random subspace method considers a subset of attributes at each node to obtain a set of trees. Random forests combine the random subspace method with bagging to build an ensemble of decision trees. They are simple to use, with two easily set parameters, and with excellent reported performance noted as the ensemble method of choice for decision trees. They are also computationally efficient and robust to noise. Various sampling approaches have been proposed in the literature, with random oversampling of minority class cases and random undersampling of majority class cases being the simplest and most common in use; others include directed sampling, sampling with generation of artificial examples of the minority class, and cluster-based sampling. A recent experimental study of various sampling procedures used with different learning algorithms found performance of sampling techniques to vary with learning algorithm

used, and also with respect to performance measures. The paper also found that simpler techniques like random over and undersampling generally perform better, and noted very good overall performance of random undersampling. Random undersampling is preferred to oversampling, especially with large data. The extent of sampling for best performance needs to be experimentally determined.

Constraint based clustering is done by modifying the objective function for evaluating clusters so that it includes satisfying constraints [79], enforcing constraints during the clustering process **,** or initializing and constraining the clustering based on labeled examples. Several adaptive distance measures have been used for semisupervised clustering, including string-edit distance trained using Expectation Maximization (EM), KL divergence trained using gradient descent, Euclidean distance modified by a shortest path algorithm, or Mahalanobis distances trained using convex optimization proposes a probabilistic model for semisupervised clustering based on Hidden Markov Random Fields (HMRFs) that provides a principled framework for incorporating supervision into prototype-based clustering. The model generalizes a previous approach that combines constraints and Euclidean distance learning, and allows the use of a broad range of clustering distortion measures, including Bregman divergences (e.g., Euclidean distance and I-divergence) and directional similarity measures (e.g., cosine similarity).

There are two major clustering techniques: “Partitioning” and “Hierarchical”. Most document clustering algorithms can be classified into these two groups. The hierarchical techniques produce a nested sequence of partition, with a single, all-inclusive cluster at the top and single clusters of individual points at the bottom. The partitioning clustering method seeks to partition a collection of documents into a set

of non-overlapping groups, so as to maximize the evaluation value of clustering. Although the hierarchical clustering technique is often portrayed as a better quality clustering approach, this technique does not contain any provision for the reallocation of entities, which may have been poorly classified in the early stages of the text analysis. Moreover, the time complexity of this approach is quadratic .In recent years, it has been recognized that the partitioned clustering technique is well suited for clustering a large document dataset due to their relatively low computational requirements .The time complexity of the partitioning technique is almost linear, which makes it widely used. The best known partitioning clustering algorithm is the K-means algorithm and its variants [4]. This algorithm is simple, straightforward and is based on the firm foundation of analysis of variances. In addition to the K-means algorithm, several algorithms, such as Genetic Algorithm (GA) and Self-Organizing Maps (SOM), have been used for document clustering. Particle Swarm Optimization (PSO) is another computational intelligence method that has already been applied to image clustering and other low dimensional datasets.

PSO was originally developed by Eberhart and Kennedy [80], and was inspired by the social behavior of a flock of birds. In the PSO algorithm, the birds in a flock are symbolically represented as particles. These particles can be considered as simple agents “flying” through a problem space. A particle’s location in the multi-dimensional problem space represents one solution for the problem. When a particle moves to a new location, a different problem solution is generated. This solution is evaluated by a fitness function that provides a quantitative value of the solution’s utility.

Merwe's research [81] indicates that utilizing the PSO algorithm's optimal ability, if given enough time, the PSO clustering algorithm could generate more compact clustering results from the low dimensional dataset than the traditional K-means clustering algorithm. However, when clustering large document datasets, the slow shift from the global searching stage to the local refining stage causes the PSO clustering algorithm to require many more iterations to converge to the optima in the refining stage than the K-means algorithm requiring. Although the PSO algorithm is inherently parallel and can be implemented using parallel hardware, such as a computer cluster, the computation requirement for clustering large document dataset is still high.

The K-means algorithm is simple, straightforward and is based on the firm foundation of analysis of variances. It clusters a group of data vectors into a predefined number of clusters. It starts with randomly initial cluster centroids and keeps reassigning the data objects in the dataset to cluster centroids based on the cluster centroid. The reassignment procedure will not stop until a convergence criterion is met.

Metalearning [74**]** is a general strategy that provides a means for combining and integrating a number of separately built classifiers or models. A metaclassifier is thus trained on the correlation of the predictions of the base classifiers. The same group has also worked on a cost-based model for fraud and intrusion detection. They use Java agents for Metalearning (JAM), which is a distributed data mining system for credit card fraud detection. A number of important performance metrics like True Positive—False Positive (TP-FP) spread and accuracy have been defined by them.

Chiu and Tsai [73] have proposed web services and data mining techniques to establish a collaborative scheme for fraud detection in the banking industry. With this scheme, participating banks share knowledge about the fraud patterns in a heterogeneous and distributed environment.

Participant Banks plays as service consumers while Fraud pattern Mining Service center (FPMSC) serves as the Service Provider. FPMSC publishes a WSDL file that describes the implementation and interface specification of its provided service.

FPMSC extracts fraud patterns from the integrated fraud transactions using Fraud Pattern Mining (FPM) algorithm. FPM algorithm is developed based on Apriori algorithm for mining fraud pattern association rules which manifest the information about what features exist in popular fraud transactions. The uncovered fraud patterns are transformed to a PMML document validated by patterns schema. The Valid PMML document is the envelope within SOAP message via the same protocols, the soap message is replied to the bank that accesses the service.

When receiving the SOAP message sent from FPMSC, participant banks can interpret the contents of PMML document within SOAP message based on patterns schema and retrieve fraud patterns. Through those fraud patterns, bank can enhance their original fraud detection systems to avoid suffering fraud attacks.

To establish a smooth channel of data exchange, Web services techniques such as XML, SOAP and WSDL are used**.** Phua et al. [8] have done an extensive survey of existing data-mining-based FDSs and published a comprehensive report. Prodromidis and Stolfo [74] use an agent-based approach with distributed learning for detecting

frauds in credit card transactions. It is based on artificial intelligence and combines inductive learning algorithms and metalearning methods for achieving higher accuracy. Phua et al. [8] suggest the use of meta-classifier similar to in fraud detection problems. They consider naïve Bayesian, C4.5, and Back Propagation neural networks as the base classifiers. A metaclassifier is used to determine which classifier should be considered based on skewness of data. Although they do not directly use credit card fraud detection as the target application, their approach is quite generic.

The problem with most of the above mentioned approaches is that they require labeled data for both genuine, as well as fraudulent transactions, to train the classifiers. Getting real-world fraud data is one of the biggest problems associated with credit card fraud detection. Also, these approaches cannot detect new kinds of frauds for which labeled data is not available. In contrast, the Hidden Markov Model (HMM)-based credit card FDS, does not require fraud signatures and yet is able to detect frauds by considering a cardholder's spending habit. Here, credit card transaction processing sequence is formed by the stochastic process of an HMM. The details of items purchased in individual transactions are usually not known to an FDS running at the bank that issues credit cards to the cardholders. This can be represented as the underlying finite Markov chain, which is not observable. The transactions can only be observed through the other stochastic process that produces the sequence of the amount of money spent in each transaction. Another important advantage of the HMM-based approach is a drastic reduction in the number of False Positives (FPs)—transactions identified as malicious by an FDS although they are actually genuine. Since the number of genuine transactions is a few orders of magnitude higher than the number of malicious transactions, an FDS should be designed in such a way that the

number of FPs is as low as possible. Otherwise, due to the “base rate fallacy” effect, bank administrators may tend to ignore the alarms.

Abhinav et al. [62] models the sequence of operations in credit card transaction processing using a Hidden Markov Model (HMM) and show how it can be used for the detection of frauds. An HMM is initially trained with the normal behavior of a cardholder. If an incoming credit card transaction is not accepted by the trained HMM with sufficiently high probability, it is considered to be fraudulent. At the same time, we try to ensure that genuine transactions are not rejected.

Outlier detection has been used for centuries to detect and, where appropriate, remove anomalous observations from data. Outliers arise due to mechanical faults, changes in system behavior, fraudulent behavior, human error, instrument error or simply through natural deviations in populations. Their detection can identify system faults and fraud before they escalate with potentially catastrophic consequences. It can identify errors and remove their contaminating effect on the data set and as such to purify the data for processing. The original outlier detection methods were arbitrary but now, principled and systematic techniques are used, drawn from the full gamut of Computer Science and Statistics.

Aggarwal [82] notes that outliers may be considered as noise points lying outside a set of defined clusters or alternatively outliers may be defined as the points that lie outside of the set of clusters but are also separated from the noise. These outliers behave differently from the normal. Particle Swarm Optimization (PSO) is swarm intelligence based metaheuristic proposed by Kennedy and Eberhart [80] which takes its inspiration from the cooperation and communication based swarm

behavior of birds, fish, bees and other insects. The approach simulates their collective effort to solve their foraging search and communication problem. High centralization, cooperation amongst the particles and simple implementation makes this method efficiently applicable to optimization problems. PSO is composed on three main components; particles, particle learning and particle movement. Particles refer to individual solutions in the solution space of a problem. Particle learning is composed of a particle's social and cognitive learning which enables the particle to move from one position to another position. The movement is supposed to be towards a better solution. Cognitive learning is represented by pBest while gBest represents social learning of the swarm. The movement of the particle is the result of the change in velocity and position of the particle.

Zhang [83] proposes a new outlier detection mechanism using the HPSO-clustering algorithm based on swarm intelligence. The approach exploits the idea of generation based swarm evolution, which uses particle swarm optimization for data clustering. HPSO-clustering uses a partitioned approach to generate a hierarchy of clusters. Each level of the hierarchy is treated as a generation of the swarm. The initial generation consists on the entire swarm. Each particle of the swarm represents a centroid of a cluster. The swarm is then evolved towards a single cluster by merging two clusters of the swarm in each successive generation.

The purpose is twofold; generating a hierarchy of clusters as well as identifying suspicious observations. A cluster based approach helps find the outliers based on their distance to the centroids. Less dense data falling at considerable distance from the centroids of the nearest cluster is considered as a potential outlier. As the outlier distance threshold of outlierness is relative, it can be different for

different data and vary from application to application. Distance is an important factor and is not easy to define as a constant for outlier detection. In the proposed approach the distance value evolves for each cluster differently. It is a product of the intra cluster distance of a specific cluster and a constant value.

Privacy preserving methods have been developed for a wide variety of data mining tasks. A generic technique for privacy preserving data mining has been proposed by creating a condensed group of data from a given data set. While forming the condensed group this method randomly chooses an object, finds its nearest (k-1) objects and then put them in the same group. But, such a method cannot be used for finding outliers, since if the random point chosen happens to be an outlier, then the method would force the nearest (k-1) objects to group with the outlier, even though these points may be very much distant from the outlier. Hence, the information about the outlier is hidden in such a group.

Challagalla [84] proposes a technique for privacy preserving outlier detection using hierarchical clustering methods. A technique for outlier detection using hierarchical clustering methods has been addressed. The use of hierarchical clustering methods in this technique is motivated by the unbalanced distribution of outliers versus “normal” cases in data sets. In almost all attempts to create the initial clusters, non-hierarchical clustering methods would spread the outliers across all clusters. Given that most of those methods strongly depend on the initialization of the clusters, and it is expected as a rather unstable approach. Therefore, the hierarchical clustering methods used, which are not dependent on the initialization of the clusters.

The clustering results of the approach are promising and the complexity of the approach is less than the traditional hierarchical agglomerative approach. This paper also highlights the scalability of the HPSO-clustering approach for outlier detection.

Some of the clustering methods such as ROCK, CLARANS, BIRCH and DBSCAN are designed to find clusters in the dataset while also finding outliers in the data but they are optimized for clustering rather than outlier detection. ROCK is proposed for clustering categorical and Boolean attributes by introducing the concept of links. In ROCK the outliers are isolated in the initial pruning phase. CLARANS motivated by PAM and CLARA employs a randomized search to find the clusters. BIRCH finds nested clustering structures in large databases. Another cluster based outlier detection method called Find CBLOF uses Clustering Based Local Outlier Factor (CBLOF) to determine if an object is an outlier

Another effective technique finds outliers based on the density of local neighborhood relying on the Local Outlier Factor (LOF) of each point, which depends on the local density of its neighborhood. In this the authors proposed a local outlier detection mechanism based on the local density of the neighborhood of the object. They compute Local Outlier Factor (LOF) for top-n outliers instead of computing for all k-neighbors.

Evaluation of the outlier detection methods is not a trivial task as the definition of the outlier itself is relative and outlier detection rules can’t be generalized for real world outlier detection. Instead the detection process relies on empirical evidence based on some rare class detection mechanism which can be derived by modifying the real world datasets.

The PAM algorithm followed by the Separation Technique (henceforth, the method will be termed PAMST). The separation of a cluster A is defined as the smallest dissimilarity between two objects; one belongs to Cluster A and the other does not. If the separation is large enough, then all objects that belong to that cluster are considered outliers. In order to detect the clustered outliers, one must vary the number k of clusters until obtaining clusters of a small size with a large separation from other clusters.

A clustering based method to detect outliers. First, the PAM algorithm is performed, producing a set of clusters and a set of medoids (cluster centers). To detect the outliers, the Absolute Distances between the Medoid, μ, of the current cluster and each one of the Points, pi, in the same cluster (i. e., $| p_i - \mu |$) are computed. The produced value is termed (ADMP). If the ADMP value is greater than a calculated threshold, T, then the point is considered an outlier; otherwise, it is not. The value of T is calculated as the average of all ADMP values of the same cluster multiplied by 1.5.

A hybrid cluster based outlier detection system, which used Partitioning Around Medoid (PAM) clustering algorithm and Absolute Distance between Medoid (ADMP) for distance based outlier detection. This method produced good results with small datasets, but the performance degraded with large datasets. The reason for this degradation while scaling up the size of dataset was because of PAM clustering algorithm. Three other algorithms, namely, CLARA (Clustering around LARge Applications), CLARANS (Clustering Large Applications based on RAndomized Search) and Fuzzy C-Means method are considered for clustering the data. After

clustering, small clusters are removed as outliers. The outliers in the large clusters are then detected.

PAM algorithm is used to identify the medoids for each of these samples. Then each object of the entire dataset is assigned to the resulting medoids. Similar to PAM, the objective function is computed to select the best set of medoids as output.

CLARANS, another portioning algorithm, is an improvement to CLARA to form clusters with minimum number of searches. CLARANS is similar to CLARA, does not check all nodes' neighbor. But, unlike CLARA, it does not restrict its search to a particular subgraph, but it searches the original graph. One key difference between CLARANS and PAM is that the former only checks a sample of the neighbors of a node. CLARA draws a sample of nodes at the beginning of a search while CLARANS draws a sample of neighbors in each step of a search. This has the benefit of not confining a search to a localized area The CLARANS procedure depends on two parameters, namely, maxneighbor and numlocal. Maxneighbor is the maximum number of neighbors examined and numlocal is the number of local minima obtained. The higher the value of maxneighbor, the closer is CLARANS to PAM, and the longer is each search of a local minima. But, the quality of such a local minima is higher and fewer local minima need to be obtained. In the procedure, nodes with progressively lower costs are searched. But, if the current node has already been compared with the maximum number of the neighbors of the node (specified by maxneighbor) and is still of the lowest cost, the current node is declared to be a "local" minimum. Then the cost of this local minimum is compared with the lowest cost obtained so far. The lower of the two costs above is stored in mincost. Algorithm

CLARANS then repeats to search for other local minima, until numlocal of them has been found.

Fuzzy c-means clustering involves two processes: the calculation of cluster centers and the assignment of points to these centers using a form of Euclidian distance. This process is repeated until the cluster centers stabilize. The algorithm is similar to kmeans clustering in many ways but it assigns a membership value to the data items for the clusters within a range of 0 to 1. So it incorporates fuzzy set's concepts of partial membership and forms overlapping clusters to support it.

A Distance Based Outlier Detection for Data Streams (DBOD-DS) is proposed by Sadik et al. [85]. DBOD-DS detects outliers based on two user-defined parameters that are neighbor radius and minimum neighbor density. But DBOD-DS is unable to handle concept evolution in streaming data. Cluster based OutlieR Minera (CORM) is a clustering-based approach for outlier detection based on k-mean. It divides data stream in chunks of data for processing. Its performance is poor on grouped outlier as it treats those as normal data clusters. Yogita et al. [86] has proposed a framework for outlier detection in evolving data streams by weighting attributes in clustering. It is a clustering based framework that assigns weights to all attributes depending on their respective relevance in clustering. Also the weights are updated periodically to adapt them according to evolving concepts.

Outlier detection in streaming data is very challenging because streaming data cannot be scanned multiple times and also new concepts may keep evolving in coming data over time. Irrelevant attributes can be termed as noisy attributes and such attributes further magnify the challenge of working with data streams. An

unsupervised outlier detection scheme is used for streaming data. This scheme is based on clustering as clustering is an unsupervised data mining task and it does not require labeled data. In proposed scheme both densities based and partitioning clustering method are combined to take advantage of both densities based and distance based outlier detection. Proposed scheme also assigns weights to attributes depending upon their respective relevance in mining task and weights are adaptive in nature. Weighted attributes are helpful to reduce or remove the effect of noisy attributes. Keeping in view the challenges of streaming data, the proposed scheme is incremental and adaptive to concept evolution.

The goal of outlier detection is to find uncommon instances in a given dataset. Outlier detection has been used in various applications such as defect detection from behavior patterns of industrial machines, intrusion detection in network systems, and topic detection in news documents. Recent studies include finding unusual patterns in time-series, discovery of spatio-temporal changes in time-evolving graphs, self-propagating worm detection in information systems, and identification of inconsistent records in construction equipment data. Since outlier detection is useful in various applications it has been an active research topic in statistics, machine learning, and data mining communities for decades.

A standard outlier detection problem falls into the category of unsupervised learning due to lack of prior knowledge on the 'anomalous data'. The list of applications that utilize outlier detection [4] are:

- **Fraud detection** - detecting fraudulent applications for credit cards, state benefits or detecting fraudulent usage of credit cards or mobile phones.

- **Loan application processing** - to detect fraudulent applications or potentially problematical customers.
- **Intrusion detection** - detecting unauthorized access in computer networks.
- **Activity monitoring** - detecting mobile phone fraud by monitoring phone activity or suspicious trades in the equity markets.
- **Network performance** - monitoring the performance of computer networks, for example to detect network bottlenecks.
- **Fault diagnosis** - monitoring processes to detect faults in motors, generators, pipelines or space instruments on space shuttles for example.
- **Structural defect detection** - monitoring manufacturing lines to detect faulty production runs for example cracked beams.
- **Satellite image analysis** - identifying novel features or misclassified features.
- **Detecting novelties in images** - for robot neotaxis or surveillance systems.
- **Motion segmentation** - detecting image features moving independently of the background.
- **Time-series monitoring** - monitoring safety critical applications such as drilling or high-speed milling.
- **Medical condition monitoring** - such as heart-rate monitors.
- **Pharmaceutical research** - identifying novel molecular structures.

- **Detecting novelty in text** - to detect the onset of news stories, for topic detection and tracking or for traders to pinpoint equity, commodities, FX trading stories, outperforming or underperforming commodities.
- **Detecting unexpected entries in databases** - for data mining to detect errors, frauds or valid but unexpected entries.
- **Detecting mislabeled data in a training data set**.

According to three fundamental approaches to the problem of outlier detection:

Type 1

Determines the outliers with no prior knowledge of the data. This is essentially a learning approach analogous to unsupervised clustering. The approach processes the data as a static distribution, pinpoints the most remote points, and flags them as potential outliers. Type 1 assumes that errors or faults are separated from the 'normal' data and will thus appear as outliers. The approach is predominantly retrospective and is analogous to a batch-processing system. It requires that all data be available before processing and that the data is static. However, once the system possesses a sufficiently large database with good coverage, then it can compare new items with the existing data.

There are two sub-techniques commonly employed, diagnosis and accommodation

- An outlier diagnostic approach highlights the potential outlying points. Once detected, the system may remove these outlier points from future processing of the data distribution. Many diagnostic approaches iteratively prune the outliers

and fit their system model to the remaining data until no more outliers are detected.

- An alternative methodology is accommodation that incorporates the outliers into the distribution model generated and employs a robust classification method.

These robust approaches can withstand outliers in the data and generally induce a boundary of normality around the majority of the data which thus represents normal behavior. In contrast, non-robust classifier methods produce representations which are skewed when outliers are left in. Non-robust methods are best suited when there are only a few outliers in the data set as they are computationally cheaper than the robust methods but a robust method must be used if there are a large number of outliers to prevent this distortion. Cheap Least Squares algorithm is applicable if there are only a few outliers but switch to a more expensive but robust algorithm for higher frequencies of outliers.

Type 2

This type models both normality and abnormality. This approach is analogous to supervised classification and requires pre-labeled data, tagged as normal or abnormal. Classifiers are best suited to static data as the classification needs to be rebuilt from first principles if the data distribution shifts unless the system uses an incremental classifier such as an evolutionary neural network. A type 2 approach can be used for on-line classification, where the classifier learns the classification model and then classifies new exemplars as and when required against the learned model. If the new exemplar lies in a region of normality it is classified as normal, otherwise it is

flagged as an outlier. Classification algorithms require a good spread of both normal and abnormal data, i.e., the data should cover the entire distribution to allow generalization by the classifier. New exemplars may then be classified correctly as classification is limited to a 'known' distribution and a new exemplar derived from a previously unseen region of the distribution may not be classified correctly unless the generalization capabilities of the underlying classification algorithm are good.

Type 3

This type models only normality or in a very few cases model abnormality**.** It is analogous to a semi-supervised recognition or detection task and can be considered semi-supervised as the normal class is taught but the algorithm learns to recognize abnormality. The approach needs pre-classified data but only learns data marked normal. It is suitable for static or dynamic data as it only teaches one class which provides the model of normality. It can learn the model incrementally as new data arrives, tuning the model to improve the fit as each new exemplar becomes available. It aims to define a boundary of normality. A type 3 system recognizes a new exemplar as normal if it lies within the boundary and recognizes the new exemplar as novel otherwise. It requires the full gamut of normality to be available for training to permit generalization. However, it requires no abnormal data for training unlike type 2. Abnormal data is often difficult to obtain or expensive in many fault detection domains such as aircraft engine monitoring. It would be extremely costly to sabotage an aircraft engine just to obtain some abnormal running data. Another problem with type 2 is it cannot always handle outliers from unexpected regions, for example, in fraud detection a new method of fraud never previously encountered or previously unseen fault in a machine may not be handled correctly by the classifier unless

generalization is very good. In this method, as long as the new fraud lies outside the boundary of normality then the system will be correctly detect the fraud. If normality shifts then the normal class modeled by the system may be shifted by re-learning the data model or shifting the model if the underlying modeling technique permits such as evolutionary neural networks.

Statistical approaches were the earliest algorithms used for outlier detection. Some of the earliest are applicable only for single dimensional data sets. In fact, many of the techniques described in both single dimensional or at best univariate. One such single dimensional method is Grubb's method (Extreme Studentized Deviate) [9] which calculates a Z value as the difference between the mean value for the attribute and the query value divided by the standard deviation for the attribute where the mean and standard deviation are calculated from all attribute values including the query value. The Z value for the query is compared with a 1% or 5% significance level. The technique requires no user parameters as all parameters are derived directly from data. However, the technique is susceptible to the number of exemplars in the data set. The higher the number of records the more statistically representative the sample is likely to be. Informal box plots also used to pinpoint outliers in both univariate and multivariate data sets. This produces a graphical representation and allows a human auditor to visually pinpoint the outlying points.

Proximity-based techniques are simple to implement and make no prior assumptions about the data distribution model. They are suitable for both type 1 and type 2 outlier detection. However, they suffer exponential computational growth as they are founded on the calculation of the distances between all records. The computational complexity is directly proportional to both the dimensionality of the

data m and the number of records n. Hence, methods such as k-nearest neighbor with $O(n^{2m})$ runtime are not feasible for high dimensionality data sets unless the running time can be improved. There are various flavors of k-Nearest Neighbor (k-NN) algorithm for outlier detection but all calculate the nearest neighbors of a record using a suitable distance calculation metric such as Euclidean distance or Mahalanobis distance.

Euclidean distance is given by

$$\sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (2.1)$$

Mahalanobis distance is given by

$$\sqrt{(x-\mu)^T}\ c^{-1}\ (x-\mu) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (2.2)$$

An optimized k-NN is used to produce a ranked list of potential outliers. A point p is an outlier if no more than n − 1 other point in the data set has a higher Dm (distance to mth neighbor) where m is a user-specified parameter. It also improved the running speed of k-NN by creating an efficient index using a computationally efficient indexing structure with linear running time.

According to the type of data distribution and the process carried out for clustering, the clustering process is divided into three types: fast clustering, hierarchy clustering and other. K-means clustering method is a classic widely used clustering method, its advantages and disadvantages are obvious. This kind of clustering method, firstly define a set of initial clustering center, then according to the input

values, assigning each record to the most similar cluster, after finish the assignment, updating the clustering center to reflect new records belong to new clusters, check again, to determine whether these records should be redistributed to different clusters, the process above will continue until achieving the maximum iteration number or the threshold value is under the standard.

But, it needs to specify the initial clustering number artificially and at the beginning of clustering process the selection of initial clustering centers has more randomness for easily producing local optimal result, and will be affected by the outliers. The influence research of data distribution on clustering results mainly focuses on the two-dimensional visualization and the description of spatial data features is insufficient.

The advantage of hierarchy clustering is not necessary to choose the initial clustering number. At the beginning, many hierarchy clustering methods make the single record as the original cluster, then recursive incorporating these records to bigger clusters. Although the obtained models in clustering process depend on the order of training data, data rearrangement could cover it.

Kohonen [87] network is a kind of executive clustering neural network type, also known as the self-organizing mapping. The basic unit is neuron, and neurons are divided into two layers: the input and output lay. All input and output neurons are connected, these connections have their weights generated randomly. In the Kohonen process, the input data will be displayed in the input layer, the value will be transmitted to the output layer, and the strongest response out neurons will be the winner and the output results, if a unit won a record, its weight will be adjusted to

match the model of predicted variables as much as possible. Kohonen network tries to reveal the model of the input fields. Usually, Kohonen network will eventually form a few collecting many observation data units (strong unit), and several no correspond to any observation data units (weak unit).

Support Vector Machine (SVM) has been widely used in many application areas of machine learning, such as text classification, handwriting recognition and intrusion detection. An extensive comparison made with SVM to 16 other classifiers. And the result showed that SVM outperformed all the other classifiers except a few cases. However, regular SVM is no longer suitable to imbalance-class especially the datasets are extremely imbalanced. An effective approach to improve the performance of SVM used in imbalanced datasets is to bias the classifier so that it pays more attention to minority samples.

This can be done by setting different misclassifying penalty. Veropoulos and Cristianini [13] have discussed this approach in their paper. A number of solutions to the class-imbalance problem were proposed both at data and algorithm level. At the data level, resampling techniques are widely used. Kubat and Matwin [33] proposed a one-sided selection process which under-sampling the majority class to remove noisy, borderline, and redundant training samples. Nitesh and Kevin et al. [31] proposed an algorithm SMOTE to over-sampling minority datasets. The advantage of SMOTE is that it makes the decision regions larger and less specific.

Various sampling strategies are comparing and analyzed the performance of these different sampling strategies. Drawback of re-sampling is that under-sampling can potentially remove certain important instances and lose some useful information,

and over-sampling can lead to over fitting. At the algorithmic level, cost-sensitive learning is an effective solution. This algorithm can improve the performance of classification by setting different misclassification cost to the majority and minority samples.

Veropoulos et al. [13] suggested using different penalty constants for different classes of data in SVM. Yuchun Tang et al. [11] discussed several SVM modeling methods for imbalanced classification and analyzed the performance. Many cost-sensitive boosting algorithms is in use. These algorithms usually raise high cost instances weights in every iteration of the boosting process. A kernel-based two-class classifier also proposed for imbalanced datasets using orthogonal forward selection. Many algorithms combining re-sampling and cost sensitive learning have also been proposed. Rehan et al .[88] combined SOMTE algorithm with cost-sensitive SVM, experiments shows that the new algorithm SDC (SMOTE with Different error Costs) performs better than SVM, US(Under-Sampling), SMOTE, and DEC(cost-sensitive SVM) in the Gmean metric**.** SMOTE Boost combined SMOTE algorithm with cost-sensitive boosting algorithm for class-imbalance learning.

One-class SVM only recognizes examples from one class rather than differentiating all examples. It is useful when examples from the target classes are rare or difficult to obtain. The classification is based on the predetermined threshold, which indicates whether to include the instance to the target class or not. In order to apply one-class SVM for multiclass problem, multiple models of one-class SVM will be trained together.

Multiple one-class SVM models as one large optimization problem to minimize the total errors, while use separate one-class SVM models and iteratively adjust the threshold of each models to maximize the total accuracy. The main challenging problem is selecting proper thresholds for each class: setting a high threshold value would make an inclusion more difficult, so positive examples are more likely to be misclassified. Yet, setting the value too low would allow more examples from non-target classes in the target class region.

2.7 Role of Collective Animal Behaviors

In many animal groups, the conflicts of interest are often inevitably generated between group members about the outcome of a consensus decision [89]. The extent of such conflicts determines the exchange of decision-related information between individuals and the degree of cooperation during decision-making. For little conflict of interest, there are some typical examples of consensus decision, such as a flock of birds navigating to a new forage patch and insects choosing a new nest site . However, many consensus decisions are more likely to involve significant conflicts of interest between at least some of group members, especially for informed individuals. This is because animal groups often have to select between mutually exclusive activities or between moving to different sites. Therefore, consensus decisions involving conflicts of interest need an underlying mechanism to reach a compromise between conflicts.

Collective behavior coordination in a network of dynamic agents has attracted a lot of attention in recent years, in particular, from physicists, biologists, mathematicians and social scientists. Investigation of the fundamental mechanisms

yielding collective behaviors, is significant not only for bio-group of animals or group activities of humans (e.g., riots, fashion and escaping panic, etc.), but also for nanotechnology applications such as spontaneous magnetization. A large volume of literature has reported some research progress in collectively migrating bacteria, insects or birds and in phenomena where groups of organisms or non-living objects reach an ordered or synchronized state such as the one corresponding to fireflies flashing in unison or people clapping in phase during rhythmic applause.

The motivation for these studies is to understand the emergence of self-organized collective behaviors in groups with inter-agent interactions. Indeed, such systems typically show interesting ordering phenomena as the individuals collectively change their behaviors to a common pattern. Dynamical systems with collective behaviors arise in biological networks at multiple levels of abstraction, from interactions among molecules and cells to the behavioral ecology of animal groups. Flocks of birds, schools of fishes and colonies of bacteria can travel information and act as one unit, allowing these creatures to exhibit complex collective behaviors such as formation-keeping during migration, obstacle avoidance, leader selection, and foraging.

During recent decades, many biologists, physicists, social experts and interdisciplinary scientists have devoted their work to investigating how flocks/swarms can exhibit high-level globally coordinated collective-behavior based on low-level distributed individual intelligence. To fulfill such a task, the first step may be solving the consensus problem, where groups of agents asymptotically agree upon certain quantities of interest like attitude, position, temperature, voltage, etc. Furthermore, exploring distributed computational methods for solving consensus

problems has direct applications to real industrial systems such as sensor network data fusion, load balancing, unmanned air vehicles (UAVs), attitude alignment of satellite clusters, and congestion control of communication networks, multi-agent formation control, and rendezvous. Among the previous works on consensus problems, the theoretical foundations of general consensus problems by investigating the relation between the eigen value distribution of the Laplacian matrix L associated with the group topology and some important consensus properties such as the achieved consensus speed and the consensus robustness to time-delays. It was also shown that a network with high algebraic connectivity is robust to both node-failures and edge-failures. To improve the speed of convergence towards consensus for homogeneous networks, murray proposed a method based on the addition of a few long links to a regular lattice, thus transforming it into a small-world network. On the other hand, for heterogeneous influence networks, by decreasing the scaling exponent in the associated power law distribution of the influence radius of each node, the ability of the network to reach direction consensus can be significantly enhanced due to the leading roles played by a few hub agents. In addition, if agents can adaptively change their velocities appropriately, the convergence speed can be remarkably improved.

Apart from investigating the consensus mechanisms for biological flocks/swarms, more and more scientists have become interested in the underlying interaction or communication mechanisms. A basic but popular flocking simulation model can be traced back to Reynolds [90], where three elementary rules are prescribed

- Separation

  Steer to avoid crowding and collision;

- Alignment

  Steer towards the average heading;

- Cohesion

  Steer to move towards the average position.

These rules have been proven effective and are often used in the design of bio-group dynamic models. In 2003, Gazi and Passino [91**]** proposed an attractive/repulsive (A/R) swarm model in which the motion of each individual is determined by two factors:

- attraction to the other individuals at long inter-individual distances;
- repulsion from the other individuals at short inter-individual distances.

With this model, they proved that the individuals typically form a bounded cohesive swarm in finite time. They later generalized their model into a social foraging swarm model by modifying the attractant/repellent profile, i.e., by additionally considering attraction towards favorable regions (or repulsion from unfavorable regions). Under some suitable circumstances, this improved model guarantees convergence to the favorable regions of the foraging profile. The A/R model of Gazi and Passino [91] has been widely adopted by physicists and biologists to mimic self-driven particles and biological swarms as it provides conditions for guaranteed cohesion of the swarm.

A very popular alignment flock model is the Vicsek model, where, at each step, every agent updates its steering direction towards the average direction of its neighbors. With the decreasing external noise or the increasing density of the agents, the collective behavior of the flock undergoes a phase transition from a disordered movement to a coherent collective movement. In Couzin [89] designed a Three-Circle model by inserting an orientation area governed by the Vicsek model between the attraction and repulsion areas of the A/R model. The corresponding Three-Circle model yields three typical types of collective behaviors, i.e., swarming, torus-shaped collective motion, and flocking. More precisely, if the internal orientation area is inexistent, the model yields a swarming behavior; if a narrow orientation area exists, a torus-shaped collective motion will occur; finally, if the orientation area is intensified to a sufficiently large size, the collective behavior will transform from a torus-shaped to a flocking motion. For its wide collective behavior coverage, this novel model has the potential to become one of the most general flocking models in the near future. Based solely on the currently available information of the network, most of the previous model analyses on flock dynamics have concentrated on properties such as congregation, stabilization, cohesion, and quick consensus.

For several decades, biologists have, however, experimentally shown that natural bio-groups possess advanced intelligence, namely predictive intelligence which endows each individual with the capability to predict the future motion of its neighbors according to their past trajectories.

In 1995, Montague et al. [92] proposed simple Hebbian learning rules to explain the predictive mechanisms in bees' foraging in uncertain environments. Other researchers focused on the predictive functions of the optical and acoustical

apparatuses of bio-groups' individuals, especially the retina and the cortex. Through extensive bio-eyesight experiments, they found that, when an individual observer prepares to eye-follow the displacement of a visual stimulus, visual adaptation is transferred from the current fixation to the future gaze position. These investigations support the conjecture of the existence of predictive mechanisms inside very many bio-groups. Based on these previous experimental results, it seems clear that decisions on the next-step behavior of each individual are not only based on the currently available state information but also on the prediction of future states. This type of behavioral decision based on prediction is used, for example, by a chameleon to capture a fly, by a dog to catch a frisbee, or by a football player to challenge for the point of first fall. This predictive mechanism reduces the negative influence of information transmission delays within flocks and facilitates the propagation of the group objective or decision information among the individuals of the group. Since nature has chosen to utilize predictive mechanisms, it is reasonable to conjecture that such mechanisms play a very important role in the emergence and evolvement of the abundant biological flocks/swarms. Figure 2.4 shows the predictive nature of flocks**.** Moreover, development of relevant analysis methods for predictive mechanism can be critical to thoroughly understand and break through the collective performance bottlenecks of industrial multi-agent systems. Collective predictive protocols are beginning to find their way into engineering areas such as autonomous robot formations, sensor networks, and UAVs. Each agent in these groups typically has limited power to send messages, and thus communication in batch mode, rather than continuous mode, is generally desirable to save energy. This is precisely one of the advantages of predictive protocols. Indeed, as will be shown in this article, predictive

protocols typically allow to sharply expand the range of feasible sampling periods and to save costly long range communications.

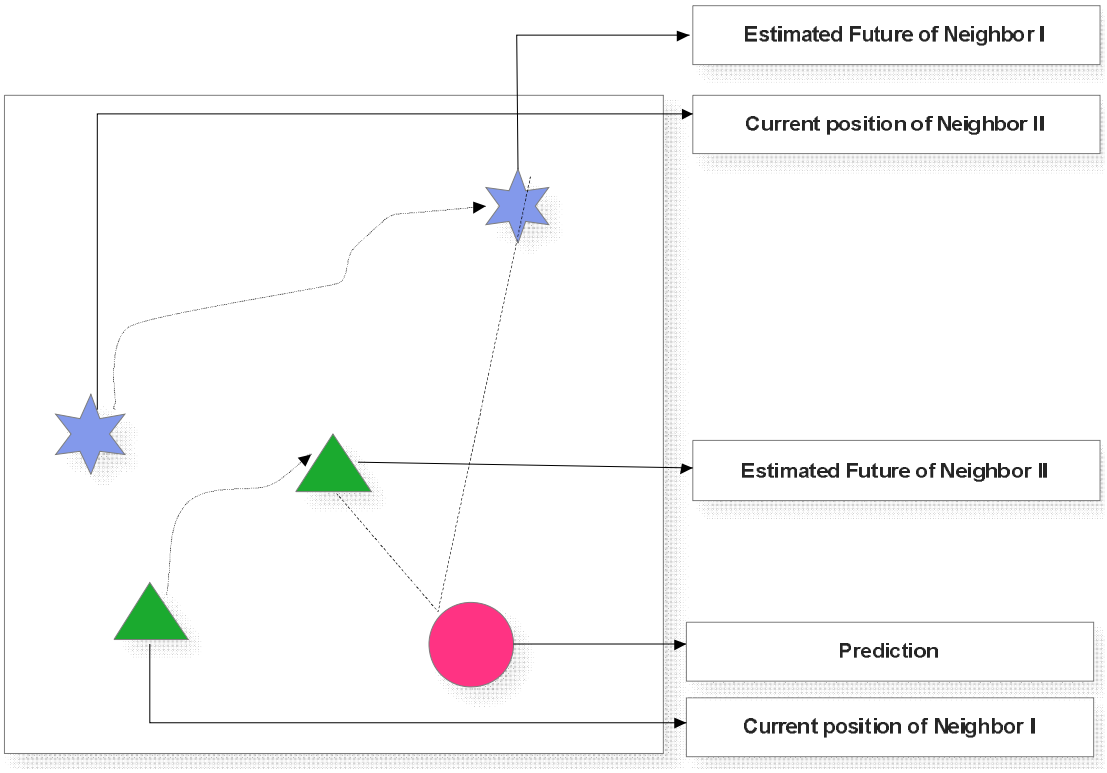

Figure 2.4 Illustration of the Predictive Nature of Flocks

A predictive protocol is designed based on the Wattz-Strogatz small-world connection model. This preliminary study shows that the intelligent predictive capability not only improves the cohesive and formative collective behavior but also reduces the long-range communication cost inside the flock. The power of predictive mechanisms is thus initially demonstrated. Kaplan has studied the dynamics of consensus, in the case of research on language, specifically the formation of a shared lexicon inside a population of autonomous agents. Language games are models of interaction among very simple agents where words "compete" to form a dominant lexicon. Agents have names for objects and they interact with each other, influencing and being influenced by others. With time, a winning option can eventually emerge,

and a consensus is reached for all the agents in the society. Social rules strike a balance between allowing agents sufficient freedom to achieve their goals and restricting them so that they do not interfere with each other. Shoham et al. [93] have investigated social rules as a design tool and have made research about the processes by which agents can make local decisions that lead to global conventions. In fact they have tested, experimentally and analytically, convergence properties of several behaviors for the on-line emergence of social conventions inside multi-agent systems. Dominance interactions are an important aspect of social behavior in many animal species. Two main theories have been suggested: dominance orders are formed due to the inherent quality of an individual or they are developed as a result of a self-organizing process. The self-organizing, model states that there are no genetic differences but it is the fight evolution that dictates the ranking. Dominance hierarchies are formed by chance and by a double reinforcement mechanism: after encounters, winners reinforce their probability of winning and losers reinforce their probability of loosing.

2.8 Summary

The literature survey provides a detailed overview of frauds that have been occurring in the banks and in specific the type of credit card frauds that are taking place. It also discusses the techniques that are proposed to overcome these frauds. The various techniques include, Artificial Neural Networks, Ant Colony Optimization Techniques, Particle Swarm Optimization, Hidden Markov Model, Support Vector Machines, Clustering and Outlier Detection Techniques. In addition to these techniques, Multi-Clustering and Collective Animal Behavior are also discussed, which is used in this study for optimizing the performance by reducing the errors.