Financial Fraud Detection: Multi-Objective Genetic Programming with Grammars and Statistical Selection Learning

Haibing Li[#1], Wing-Lun Lam[#2], Chi-Wai Chung[#3], Man-Leung Wong[#4]

[#]*Department of Computing and Decision Sciences*
Lingnan University, Hong Kong

Abstract

Financial fraud is a serious problem that often produces destructive results in the world and it is exacerbating swiftly in many countries. It refers to many activities including credit card fraud, money laundering, insurance fraud, corporate fraud, etc. The major consequences of financial fraud are loss of billions of dollars each year, investor confidence and corporate reputation. Therefore, a research area called Financial Fraud Detection (FFD) is obligatory, in order to prevent the destructive results caused by financial fraud. In this study, we propose a new approach based on multi-objectives optimization, Genetic Programming (GP), grammars, and ensemble learning for solving FFD problems. We comprehensively compare the proposed approach with Logistic Regression, Neural Networks, Support Vector Machine, Bayesian Networks, Decision Trees, AdaBoost, Bagging and LogitBoost on four FFD datasets including two real-life datasets. The experimental results showed the effectiveness of the new approach. It outperforms existing data mining methods in different aspects. There are two major contributions of the study. First, it evaluates a number of existing data mining techniques on the given FFD problems. Second, it suggests a new approach for handling these far-reaching problems. Moreover, a novel ensemble learning method called Statistical Selection Learning is proposed.

Keywords — *Financial Fraud Detection, Multi-objective Optimization, Grammar-Based Genetic Programming, Ensemble Learning.*

I. INTRODUCTION

Financial fraud is a serious problem that often produces destructive results in the world and it is exacerbating swiftly in many countries, such as China. It is a criminal act, which violates the law to gain unauthorized financial benefit [1]. Financial fraud refers to many activities, such as credit card fraud, money laundering, insurance fraud, corporate fraud, etc. Credit card fraud and corporate fraud have attracted a great deal of attention from the year of 1998, and are still in the trend of escalation [1]. Credit card fraud is about unauthorized usage of a credit card, unusual transaction behaviour or transactions on an inactive card [2], [3]. In the era of rapid development of information technology, a vast volume of information can be created every second, but there can be a lack of powerful techniques that can analyze the information. It is costly to detect the potential fraudulent transactions manually. The results may be destructive if one chooses to ignore them or detect them incorrectly. At the same time, credit cards are the most popular transaction method with increasing users, but the credit card fraud rate is also increasing.

Corporate securities fraud is related to corporate fraud in listed firms. For example, it may be perpetrated to increase the stock prices of fraudulent firms, to obtain more loans from banks or repay lesser dividends to shareholders [4]. In the U.S., financial analysts have been confirmed to contribute to corporate fraud detection. Effective external monitoring can increase the confidence of shareholders or investors, which is crucial to the functioning of any capital market [5]. It is also important for China's securities market, as corporate fraud can impede China's economic development since it has serious consequences for shareholders, employees and society [5]. No matter what type of fraud is involved, it results in losses of billions of dollars every year [6]. Since the amount of fraud is increasing rapidly, the workload of auditors is also increasing. They have become overburdened with the task of detection of fraud. Various efficient financial fraud detection techniques are required to detect which ones will commit a fraud.

Financial Fraud Detection (FFD) is vital to prevent the destructive consequences of financial fraud. It can distinguish fraudulent information from data, thereby discovering fraudulent activities or behaviour and enabling decision makers to develop appropriate policies and strategies to decrease the influences of fraud [1]. In recent years, FFD has become a hot spot domain, because it has many features like other Data Mining (DM) problems. DM plays an important role in FFD, since it uses model(s) to automatically discover useful patterns from massive data repositories [7], [8]. DM is also called Knowledge Discovery in Database (KDD), which is a process of knowledge extraction. [9] defined DM as a process of identifying interesting patterns in datasets, which can

be used in decision-making. [10] specified DM as a process that uses some techniques such as statistical, mathematical, artificial intelligence and machine learning to extract useful information from datasets. [11] stated that fraud detection has become one of the best real-world applications of data mining in industry and government. Thus we are interested in evaluating different DM methods on various FFD problems and developing better approaches for handling these problems.

In this study, four financial fraud datasets are used. Two of them are benchmark datasets about credit card fraud, and collected from UCI machine learning repository. The other two are real-life problems: U.S. Corporate Securities Fraud (U.S. CSF) and China Corporate Securities Fraud (CCSF). In order to solve these FFD problems, it is necessary to consider the kind of knowledge to be extracted from these datasets. In finance, researchers usually find different information and observations from the data to examine hypotheses about the factors in relation to committing financial fraud, such as corporate governance [12], [13], [14], investor beliefs about industry business conditions in regard to initial public offerings (IPO) [15] and impact of enforcement actions [16]. In general, these researchers prefer to apply traditional statistical methods, such as logistic regression, to achieve their goals.

On the other hand, researchers in data mining apply different data mining techniques to find out the patterns that can be used to explain the reasons for financial fraud, and then use the discovered patterns to prevent and predict financial fraud. The main difference between different data mining studies on FFD problems is that they apply or propose different data mining techniques to learn models from different FFD problems and evaluate their classification ability. Existing popular data mining techniques on FFD problems are decision trees [17], [18], neural networks [19], [17], [20], [21], support vector machines, Bayesian networks [18], [17] as well as different ensemble learning algorithms including Bagging [22], LogitBoost and AdaBoost [23].

In this study, we employ the classification accuracies on fraudulent and non-fraudulent cases as the evaluation criteria to compare the performance of the above methods. The experiment results provide a general overview of the performance of these techniques. However, it is observed that they cannot handle the FFD problems very well, especially on the two real-life datasets. Therefore we pursuit to develop new methods that can outperform the existing approaches for the FFD problems.

Recent evidence from the literature on data mining shows that some variants of Evolutionary Algorithms (EAs) are promising [24], [25], [26], [27], [28], [29]. Grammar-based Genetic Programming (GBGP) is one of the most appropriate methods among the variants of EAs [30], [31], because it can generate a set of classification rules to represent knowledge learnt from the dataset. Compared with other data mining techniques, such as neural networks, the generated classification rules are more understandable for general users. Thus we also evaluate GBGP on the FFD datasets. The experiments show that it can produce competitive results among the existing methods and it still has much room for improvement.

For FFD problems, decision makers want to find accurate, general, understandable, and interesting classification rules or patterns from the datasets. However, the original GBGP cannot handle problems with multiple objectives (e.g. accurate versus general). In order to learn better classification rules, multi-objective optimization methods [32], [25], [26], [33], [28] are integrated with GBGP to produce a set of non-dominated classification rules on all objectives. A novel ensemble learning method is then used to select rules to form an ensemble of classification rules. The proposed new approach is evaluated on the four FFD datasets and it is observed that the new approach outperforms existing data mining methods and the original GBGP in different aspects.

The rest of this paper is organized as follows. In Section 2, the background and literature review of this work are discussed. In Section 3, the proposed approach is described in detail. The motivations of different techniques and the framework of the proposed approach are discussed. In Section 4, a number of experiments are conducted to compare the performance of the proposed approach with other data mining methods. The experiment results are presented and discussed comprehensively in this section. The major findings, contributions, and business implications of this study are discussed in the last section.

II. BACKGROUND

A. *Financial Fraud*

[34] defined financial fraud as “a deliberate act that is contrary to law, rule, or policy with intent to obtain unauthorized financial benefit”. It has always been a very important research topic, and also has attracted a lot of concern. As an increasingly serious problem, financial fraud results in the loss of billions of dollars each year [6]; therefore financial fraud detection (FFD) is required in order to prevent the destructive results caused by financial fraud. FFD has many common features like other data mining problems. It has drawn a lot of research interest and a number of different techniques from many areas have been applied to tackle this problem. Especially in the field of artificial intelligence, a number of novel and advanced approaches have been developed in financial fraud detection. [1] summarized 49 journal articles on the subject published between 1997 and 2008, and found credit card fraud (14.35%) and corporate securities fraud (34.7%) have attracted a

great deal of attention during that period, and are still escalating. In this study credit card fraud and corporate securities fraud are investigated comprehensively

1) Credit Card Fraud:

Credit card fraud concerns the illegal usage of credit cards, such as unusual transactions [2]. It is difficult to determine the level of credit card fraud, since banks and companies are reluctant to release fraud figures to the public and these figures are growing over time [35]. Although these companies and banks lose billions of dollars every year due to credit card fraud, identifying which customers are included in the fraud figures is a complicated task [35]. With the constant rise of people's consumption standard, the number of users of credit cards is also increasing rapidly. At the same time, with credit cards being the most popular transaction method, the number of credit card frauds is also increasing. Detecting credit card fraud has drawn a lot of research interest and many different advanced techniques have been developed [36].

2) Corporate Securities Fraud:

Corporate securities fraud in this study is close to corporate fraud in listed companies, rather than securities fraud only, since the definition of securities frauds includes someone manipulating the securities market, modifying securities accounts or committing wire fraud [37]. On the other hand, corporate securities fraud is related to falsification of financial reports, self-dealing by corporate insiders and hiding important information from stakeholders [38]. In other words, corporate securities fraud is closely associated with their own inside problems.

In the U.S., financial analysts have been confirmed to contribute to corporate fraud detection. Effective external monitoring can increase investors' confidence, which is crucial to the functioning of any capital market [5]. It is also important for China's securities market, as corporate fraud can impede China's economic development since it has serious consequences for stakeholders, employees and society [5]. In recent years, corporate securities fraud detection has become a hot spot domain in finance and there is a wave of research papers that have studied effective policies to detect and reduce fraud.

In China, the Securities Regulatory Commission (CSRC) serves as the main regulator of securities markets in China, which is devoted to investigating the potential violations of securities regulations and instigate different enforcement actions on those fraudulent corporations that have violated the related laws. Any of the enforcement actions by CSRC will affect the stock price of the firm, even resulting in bankruptcy [12].

Prior studies on the causes of corporate securities fraud have focused on different types of determinants, such as agency problems, business pressures and corporate governance [13], [15], [39]. [40] investigated the relationship between corporate lobbying and fraud detection. They used lobbying expenses as the learning data, and found that the corporate lobbying could be an important factor in detecting corporate fraud. That is, most fraudulent firms have higher lobbying expenses than non-fraudulent firms. [41] deeply analyzed the corporate governance system of many U.S. firms and found Securities and Exchange Commission (SEC) played a very minor role in the discovery process, but analysts, employees and newspapers have strong roles to play in determining whether a firm will commit fraud or not. [14] discussed the relation between the corporate governance of a firm and information disclosure. The most important finding of the study was that larger firms adopt stricter disclosure rules than smaller firms, and firms with better disclosure rules have capable employees at management level. Moreover, firms with better disclosure rules can probably reduce the incidence rate of outright fraud by insiders.

In China, it is also necessary to verify whether larger firms will have less enforcement actions by CSRC or not. In addition, [12] examined these enforcement actions to explain whether the ownership and governance structures of corporations have impacts on committing fraud. The authors concluded that the proportion of outside directors, the tenure of the chairman and number of board meetings are factors related to committing fraud.

[42] examined the association between the financial reporting system and the quality of the corporate governance system. They considered the board members, number of financial experts and number of board meetings in the firm. As found in prior research, poor governance occurs in fraudulent firms. [16] investigated enforcement actions from the viewpoint of fraudulent firms rather than what factors lead to fraud. They found that many of these firms have problems with published financial statements and irregular reports, such as inflated profit, false statements and major failure to disclose information, which are the common problems identified by CSRC.

Considering the laws on federal securities, [43] examined the four attributes that might associate with fraud including the number of defrauded investors, assets size, losses and financial distress of the firm. The authors concluded that only financial distress has a significant impact on the presence or absence of an enforcement action. In general, since the result of the enforcement action is either yes or no, it is more reasonable to use a bivariate probit model as the learning method to analyze the data.

There is a large dataset on China's listed firms collected based on the above studies and findings for this research, in order to find out corresponding relationships to detect whether a company is fraudulent or non-fraudulent in China.

B. Data Mining Techniques in Financial Fraud Detection

The probit model, logistic regression and their variants are the most popular methods used by financial researchers [12], [13], [15], [39], [41], [14], [16], [43]. In addition to traditional statistical approaches, there is a number of machine learning techniques applied in solving financial fraud detection problems. Except for regression, neural networks might be the first and also the most popular machine-learning technique used for solving different real-world problems. [44] applied neural networks for credit card fraud detection. Instead of using resampling techniques for the given unbalanced fraud data, the authors devoted themselves to increasing the inherent correct diagnosis for legal cases from 99.9% to 99.955%. However, this performance measurement may not be the best choice for fraud detection since the accuracy always biased to the majority class (i.e. the class with a higher number of instances). [45] investigated the efficacy of using decision trees, neural networks and logistic regression for credit card fraud detection problems in order to reduce banks' risk. Moreover, the authors found that the conventional neural networks and logistic regression approaches obtained better results than decision trees.

[46] used a multi-classifier meta-learning approach for real-world credit card fraud detection. The approach is based on creating data subsets with appropriate class distribution, since most fraud detection problems have unbalanced class distributions. Moreover, they applied four learning algorithms (C4.5, CART, RIPPER and BAYES) as the base-level classifiers, and applied BAYES to train the base-level classifiers to generate the final ensemble model. The proposed method could handle the learning tasks efficiently. [3] comprehensively evaluated and compared support vector machines, random forests, and logistic regression for detecting credit card fraud on real-life datasets.

In [36], frauds are detected by using a Hidden Markov Model (HMM) modeling the sequence of actions in credit card transactions. The case is fraudulent if the incoming credit card transaction is rejected by the trained HMM with a threshold probability. The HMM is able to outperform other conventional techniques, such as neural networks, meta-learning method and Naive Bayesian networks. Since most fraud detection problems have imbalanced class distributions, [47] evaluated the performance of using the undersampling method, Synthetic Minority Over-sampling Technique (SMOTE) and EasyEnsemble to solve the problem in the unbalanced dataset, and the latter two methods returned higher accuracies than the undersampling method. Furthermore, the authors suggested three incremental learning approaches, called the "static", " update" and "forget" approaches. The best results were obtained by using the "forget" approach as the incremental learning method with EasyEnsemble.

Association rules are also useful classifiers in solving the credit card fraud detection problem. The performance measurement of each rule is determined by a confidence and support framework. A rule with high support value and high confidence value will be ranked as the top level. By using this framework, the results are more straightforward in facilitating fraud analysis [48]. [11] discussed and compared the performance of a number of different machine learning techniques, such as fuzzy logic, genetic programming and neural networks that are used in credit card fraud detection, and pointed out the advantages of these techniques based on different criteria.

[49] applied traditional logistic regression on financially fraudulent listed firms in China from 2002 to 2004. The major findings of the study are useful for corporations and auditors to detect financial frauds. For example, there is a negative relationship between the market competition and the probability of the firm to commit fraud in their financial statements.

[50] evaluated the performance of a multi-criteria decision aid classification tool in detecting the financial problems from financial statements of listed firms in Greece. They selected variables that are often used in the falsified financial statements, such as the ratio of total debt to total assets, sales ratio and net profit. The multi-criteria decision aid classification tool is able to obtain high accuracy in estimating the probability that occurs in fraudulent firms.

[17] evaluated the effectiveness of decision trees, neural networks and Bayesian belief networks in detecting and identifying the factors associated with fraudulent financial statements (FFS). In terms of their performance, Bayesian belief networks outperform others in regard to accuracy rate. An improved version of neural networks with fuzzy logic is presented in [19]. The data were collected from the U.S. Securities and Exchange Commission (SEC) between 1980 and 1995 with enforcement actions, but the size was small (i.e. 200 cases in total) and out of date. However, the proposed method outperforms most traditional statistical models and neural networks in previous studies in classifying fraudulent cases. [51] examined the effectiveness and limitations of different data mining techniques for identifying financial statement fraud. They explored a self-adaptive framework with domain knowledge to detect fraud.

[52] evaluated the financial problems in China by using classification and regression trees (CART), and compared it with traditional Logit regression. In their study, an improved data representation is introduced to describe each data item easily, and the proposed version of CART produces better results and outperforms Logit

regression. [18] highlighted the importance of a number of financial ratios, which can significantly determine the classification results. They applied an ensemble method called Stacking that combines a number of different base-level classifiers to generate an ensemble model. The performance of the ensemble model is better than that of the base-level classifiers.

C. Multi-Objective Optimization Problems

Many real-world problems can be regarded as multi-objective optimization problems and it is usually difficult to obtain a single solution for these problems. For example, it is not difficult for a stock buyer to choose a stock with the highest expected return. However, if the buyer is also concerned about the risk of the selected stock, the problem becomes more complicated because there is a relationship between expected return and risk. The stock selected by the buyer may also have very high risk, thus the buyer may consider other alternatives with smaller expected return and risk. In general, researchers search for a single solution by assigning different weights for each objective. The weight is used to describe the importance of the objective. For example, if users want to optimize Equation (1)

$$y = Maximize(w_1 * objective_1 + w_2 * objective_2) \quad (1)$$

where w_1 is the weight for $objective_1$, w_2 is the weight for $objective_2$, and $w_1 + w_2 = 1$, the first issue is to assign the value for each weight. If the users think $objective_1$ is more important than the other, then the weight of w_1 should be higher than w_2. However, it is not easy to determine the right values for these weights and the solutions will be significantly deteriorated if inappropriate weight values are used. Moreover, if there are more objectives, the problem of assigning weight values becomes even more challenging. Therefore, methods to solve multi-objectives optimization problems are required.

The main purpose of these methods is to find a number of trade-off solutions (i.e. Pareto solutions or non-dominated solutions), which can meet all objectives, and then the users can make a final decision based on the optimal solutions obtained [25]. The optimal solutions form a curve, called Pareto front, among all objectives, which is shown in Figure 1.

Without loss of generality, we define multi-objective maximization problems as follows.

$$Maximize\ f(x) = [f_1(x), f_2(x) \ldots, f_k(x)] \quad (2)$$

subject to

$$g_i(x) \leq 0,\ i = 1, 2, \ldots, m \quad (3)$$

$$h_j(x) = 0,\ j = 1, 2, \ldots, p \quad (4)$$

where $x = [x_1, x_2, \ldots, x_n]$ is a vector of $_n$ decision variables, $f(x) = [f_1(x), f_2(x), \ldots, f_k(x)]$ is called objective functions, and g_i, h_j , $i = 1, 2, ..., m$, and $j = 1, 2, ..., p$ are the constraint functions of the problem. The number of p (i.e. equality constraints) must be smaller than the number of n (i.e. objectives); otherwise the problem is over-constrained (i.e. no solutions) [25]. A Pareto or non-dominated solution x satisfies all constraint functions and there exists no other feasible solution x' which would increase some objective values without causing a simultaneous decrease in at least one of the other objectives. These Pareto or non-dominated solutions are good trade-offs for the multi-objective optimization problem.

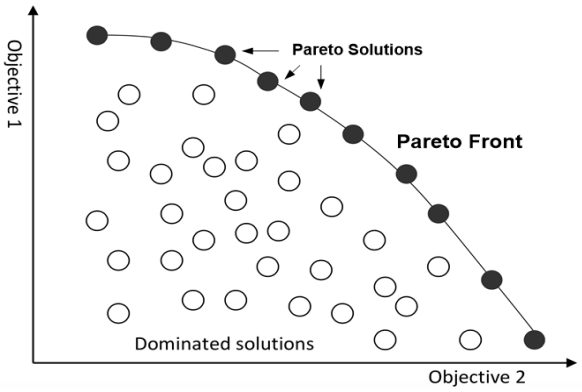

Fig 1: Example of Pareto Front

D. Genetic Programming (GP)

Genetic Programming (GP) is a population-based optimization method that extends traditional genetic algorithms [53], [54] to automatically induce computer programs [55], [56]. Unlike Genetic Algorithms (GA), GP uses a tree structure to represent an individual in a population. The overall evolutionary process of GP is depicted in Figure 2. Firstly, GP randomly creates a number of computer programs (i.e. individuals), which are composed of functions and terminals to form an initial population in generation 0 [57]. Some possible computer programs with a function set and terminal set are shown in Figure 3. For these programs, the terminal set is {0, 1, 2, -1, -2, x}, and the value of each terminal node is selected from the terminal set, which is located in the leaves of individual program trees. For example, in Figure 3(a), x,1 and 0 are the terminal nodes. The function set is {+, −, *}, and the value of a function node is selected from the function set, which is the connector of some other nodes, such as − and + in Figure 3(a).

Afterwards, it iteratively selects some individuals based on their fitness values and breeds them into a new generation of individuals by using different genetic operators. Fitness value is the score to measure the quality of the individual, which means that a good computer program has a high fitness value. The selection is based on fitness values; therefore poor computer programs have lower probability of being selected. When one or two individuals are selected, genetic operators will be applied to produce new individuals. Genetic

operators include crossover, mutation and reproduction.

The new individuals (i.e. offspring) will replace some individuals (i.e. parent) according to the altering scheme at each generation. In general, poor individuals have high probability of being replaced by new individuals. The evolution is repeated until the termination criterion, such as the number of iterations executed, is satisfied. Finally, the evolved population contains a number of good individuals to solve the given problem [55]. In order to apply GP for a problem, the user needs to specify a set of primitive functions **F**, a set of terminals **T**, a fitness function, a set of related parameters for evolution (e.g. crossover rate, mutation rate and selection rate) and the termination criteria [29]. [27] proposed a multi-objective genetic programming system to find decision trees for cost sensitive classification problems.

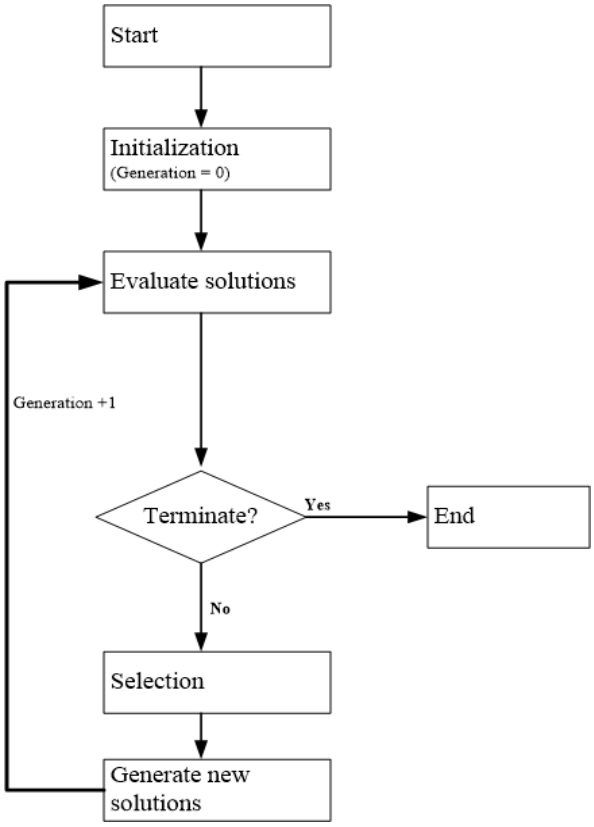

Fig 2: General process of Genetic Programming (GP)

E. Ensemble Learning

It becomes more and more difficult to improve the performance of a single classifier significantly. Moreover, it is also not sufficient to apply a single classifier to handle a difficult problem. Ensemble approaches are learning algorithms that select and combine a set of classifiers for classifying the unknown data [58]. The ensemble techniques are proven empirically and theoretically to give better results than any single classifier in most cases [59]. There are two levels of learning to generate an ensemble. The first level is called base-level, where the base learning algorithms are applied to learn base-level classifiers from the training data. The second level is called meta-level, where an algorithm is used to combine the outputs from the base-level classifiers. If the classifiers in an ensemble are trained by the same algorithm (e.g. neural networks), then it is called a homogeneous ensemble. Popular homogeneous ensemble learning techniques includes Bagging [22] and Boosting [23]. If the classifiers in an ensemble are trained by different algorithms (e.g. neural networks and decision trees), then it is called a heterogeneous ensemble. The popular heterogeneous ensemble methods include Stacking [60].

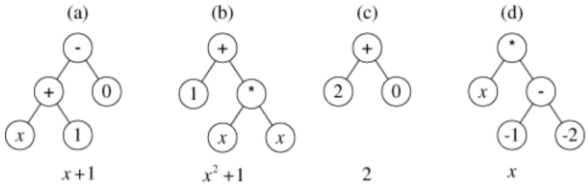

Fig 3: Four individuals in GP

III. GRAMMAR-BASED MULTI-OBJECTIVES GENETIC PROGRAMMING WITH ENSEMBLE

The general framework of the proposed method for solving financial fraud problems is shown in Table I. Three major components are included in this framework. The first consists of Grammar-based Multi-objective Genetic Programming (GBMGP), which is described in Section III-A. The second consists of ensemble learning, which is described in Section III-B. The third consists of minority prediction in model testing, which is discussed in Section III-C.

TABLE I
General framework of using GBMGP with an Ensemble Learning technique

The input to the system:
- Datasets: Training and testing datasets
- Objectives: A number of objectives, and maximization or minimization of each objective
- Pre-defined grammar for the specific problem.
- Parameters for evolution: Number of generations and number of individuals.
- - Ensemble learning technique.

1. Grammar-based Multi-objective Genetic Programming (GBMGP):
   - Applying genetic programming to learn classification rules from the training dataset.
   - Training is guided by the pre-defined grammar.
   - Output the population with evolved classification rules.
2. Generating ensemble:
   - Applying ensemble approach for the population.
3. Testing:
   - Applying the final ensemble on the testing dataset.

A. Grammar-Based Multi-objective Genetic Programming (GBMGP)

This subsection describes the key components of Grammar-Based Multi-objective Genetic Programming (GBMGP) and the corresponding motivations, designs and implementations in detail.

Figure 4 shows the general process of GBMGP. Compared with traditional Genetic Programming [55], GBMGP has three more components, which are Step 4, Step 5 and the Grammar, to handle multi-objective problems, maintain the diversity of classification rules and guide the evolutionary process respectively. The well-known multi-objective learning algorithm called Non-dominated Sorting Genetic Algorithm II (NSGAII) is applied in Step 4 [28]. A diversity maintenance scheme called Token Competition is applied in Step 5 [29]. Section III-A1 shows how Grammar-Based Genetic Programming (GBGP) [29] is used in the entire evolutionary process. Section III-A2 elaborates the multi-objective approach.

1) Grammar-based Genetic Programming (GBGP):

Comparing GBGP [30] with traditional Genetic Programming (GP), the concept of grammar is employed, which is used to control the structures evolved during the evolutionary process. GBGP supports logic grammars, context-free grammars (CFGs) and context-sensitive grammars [61] to generate tree-based programs. A suitable grammar is designed for solving a particular problem.

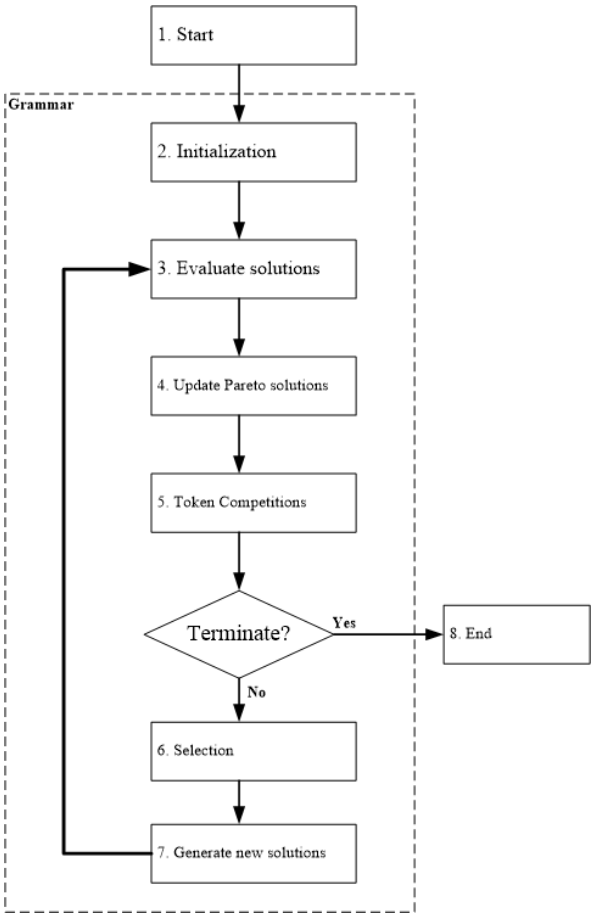

Fig 4: General process of GBMGP

Table II shows an example of a simple grammar. The genetic operations (e.g. crossover and mutation) will be executed based on the grammar, so that the new offspring generated must be valid according to the grammar.

For example, consider two individuals shown in Figure 5, which are generated based on the grammar in Table II. Individual (a) indicates that if the number of board meetings is greater than 40, then the If-Exp expression returns "yes" (i.e. the firm is fraudulent); otherwise it returns "no" (i.e. the firm is not fraudulent). Individual (b) indicates that if the number of board members is smaller than 15, then the If-Exp expression returns "yes"; otherwise it returns "no".

TABLE III
Example of a grammar to control the evolutionary process

If-Exp → *Boolean-Exp Then Else*
Boolean-Exp → *Operator Term Value*
Boolean-Exp → true | false
Term → meeting | board
Value → [0,100]
Operator → = | >= | <= | ▷ | <
Then → yes
Else → no

GBGP can learn programs in various programming languages and induce knowledge in different representations such as fuzzy Petri nets and first-order logical relations [31]. The system is also powerful enough to represent context-sensitive information and domain dependent knowledge. This knowledge can be used to accelerate the learning speed and/or improve the quality of the induced programs and knowledge [29].

The original GBGP has been applied in classification rule learning. A classification rule is a statement in the format of "If antecedents then consequent", which is commonly used by human to represent knowledge. Classification rule learning tries to learn rules from a dataset. In GBGP, a tree structure is used to represent a rule and grammars for rules have been developed to create the appropriate tree structures. A fitness function has been used to evaluate the evolved rules. The fitness function applies the support-confident framework proposed by [62]. In this framework, two objectives, *support* and *confidence*, are considered at the same time.

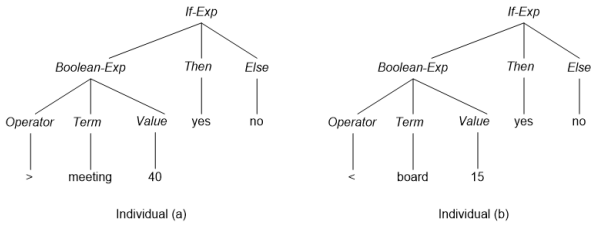

Fig 5: Example of two individuals

Support is used to evaluate the coverage of a rule. A good rule should have an appropriate support

value so that it covers a number of cases of a dataset. Support is a ratio of the number of cases matching both the *antecedents* and the *consequent* to the total number of cases.

Confidence measures accuracy. It is a ratio of the number of cases matching both the *antecedents* and the *consequent* to the number of cases fulfilling the *antecedents* only.

GBGP evaluates the support and confidence values of each rule and calculates the *normalized confidence* which is *confidence* * *log*(*confidence*/*prob*) where *prob* is the ratio of the number of cases matching the consequent to the number of cases. Since there are two different objective values but GBGP has only one fitness function, GBGP combines the two objective values into a single fitness value by using the following equation,

$$fitness = w_1 * support + w_2 * normalized\ confidence \quad (5)$$

where w_1 and w_2 are weights to control the balance between support and normalized confidence. GBGP has been applied to learn rules from medical datasets and good performance has been obtained [29].

2) Multi-Objective Evolutionary Algorithms:

As we have discussed in the previous subsection, each rule is evaluated by using two objectives. However the two objectives are conflicting and thus a good rule may be eliminated if we simply combine the two objective values of a rule to be its fitness value. For example, consider a dataset containing 50 positive and 50 negative cases. A general rule classifying all cases as positive has a support value of 0.5 and a confidence value of 0.5. A more specific rule classifying 10 positive cases correctly has a support value 0f 0.1 and a confidence value 0f 1.0. Depending on the exact values of w_1 and w_2, the second rule may be eliminated during the evolution process even though it is a good rule, because the fitness value of the second rule is smaller than that of the first rule.

However, it is not easy to determine the right values of w_1 and w_2 so that good rules can be maintained. Multi-objective optimization methods can thus be used to keep these good rules. The main idea of multi-objective optimization methods is to obtain a number of non-dominated solutions, which are also called Pareto solutions. The general background of multi-objective optimization problems has been discussed in Section II-C. As shown in Figure 6, the non-dominated solutions (i.e. black points) will form a curve called a Pareto front. These non-dominated solutions should be good classification rules. A number of Multi-objective Evolutionary Algorithms (MOEAs) can be applied and we use Non-dominated Sorting Genetic Algorithm II (NSGA-II) because it has been proven to be a powerful, fast and efficient algorithm [28].

There are two main features in NSGA-II [28], as shown in Figure 6. The first feature is to sort the individuals into different level of fronts (i.e. ranks). The individuals in the first front are the non-dominated solutions (i.e. Pareto solutions). The individuals in subsequent ranks are poorer than the individuals in previous ranks. The second feature is to measure the crowding distance between an individual and its neighbours. Incorporating NSGA-II with GBGP, each classification rule is sorted into different ranks and its crowding distance is obtained. Firstly, all non-dominated rules will be assigned the highest rank. The crowding distance of a non-dominated rule is the size of the largest cuboid enclosing it without including any other non-dominated rules. In order to find the ranks and the crowding distances of other rules, the non-dominated rules are assumed to be removed from the population and thus another set of non-dominated rules can be obtained. The ranks of these rules should be smaller than those of the previous non-dominated rules. The crowding distances of them can also be found. Similarly, the same approach can be applied to find the ranks and the crowding distances of all other rules. After finding the ranks and crowding distances, the classification rules that are located in less dense regions (i.e. those rules with large crowding distances) and have high ranking values will have higher probability of being selected as parents, and GBGP will produce new classification rules based on the selected individuals.

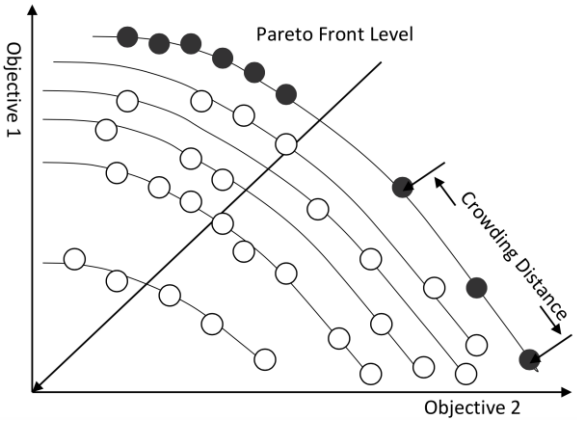

Fig 6: Example of a population sorted by NSGA-II

B. Statistical Selection Learning

As mentioned in Section II-E, ensemble learning is able to improve the performance of the classifiers. In this study, a number of classification rules are evolved by GBMGP. It is difficult to determine which of them should be selected eventually. If all of them are selected as members of an ensemble, the classification performance of the ensemble may be deteriorated, because some poor individuals may produce incorrect results. In general, the performance could be improved if an appropriate ensemble method is used. Therefore, we adopt the

ideas from ensemble learning and develop a novel ensemble technique for solving the FFD problems.

The new method selects and combines a number of evolved rules into an ensemble for achieving two different goals. The first one is to improve the classification performance of the final ensemble. The second is to accomplish diversity maintenance, which aims to have a wide variety of rules to cover more cases. From the point view of statistics, a diverse population is composed of a number of different small groups of individuals, which are significantly different from each other. Therefore, we propose an ensemble method called statistical selection learning (SSL). Suppose there is a population with different evolved individuals, and each individual *i* contains two terms, fit_i and s_i, where fit_i is the fitness value of the individual. Fitness value is calculated by finding the average of support and confidence. s_i is the status of the individual *i*, which indicates if it is selected or not. At the beginning, a set of individuals with small size (e.g. 3) is randomly selected from the Pareto front as the primary set, and then the same number of individuals are randomly selected from the whole population as the secondary set. We determine if the two sets are different by using t-test. The two sets are merged to form a new primary set if they are significantly different from each other at the 5 percent significance level. Then the above steps are repeated to compare the new primary set with another secondary set. On the other hand, the secondary set is reselected if it is not significantly different from the primary set. Once the termination conditions are satisfied, the final ensemble is constructed. The whole process of the GBMGP is shown in Algorithm 1.

C. Minority Prediction

The last step of the proposed framework is to evaluate the constructed ensemble on the testing dataset. In general, a testing case will be evaluated only if the values of attributes from the testing case are covered by the antecedent part of a rule. Otherwise, the testing case will skip the rule, and keep looking for other rules in turn until all rules are considered. However, there is no prediction for the testing case if no rules can be applied for it. Therefore, we suggest setting a default rule that classifies such testing case as belonging to the minority class (i.e. fraudulent). If the real class of the testing case is the same as the minority class (i.e. the testing case is a fraudulent firm), then we say it is correctly classified. Otherwise, it is misclassified. Therefore, minority prediction should improve the performance in classifying the fraudulent class. In this study, we think the detection of minority class (i.e. fraudulent) is much more important than the detection of majority class (i.e. non-fraudulent) in the given FFD problems. For example, if a firm is fraudulent, and it is incorrectly classified as non-fraudulent, then the loss to relative people (e.g. shareholders) may be destructive. However, if a firm is non-fraudulent, and it is incorrectly classified as fraudulent, it may need to be investigated by the China Securities Regulatory Commission (CSRC) or Securities and Exchange Commission (SEC) at relatively much lower cost (i.e. investigation fees) compared to the destructive consequence caused by fraud without any investigation. Therefore, it is more important to classify the fraudulent firms correctly than non-fraudulent firms.

Algorithm 1 GBMGP with Statistical Selection Learning

Define R_t, P_t, Q_t, E_t {The whole, parent, offspring and ensemble populations}
Define N {The population siz}
set $t = 0$ {The initial generation.}
for i = 1 to N **do**
  Initialize and evaluate the individual ind_t. Store it into P_t.
while terminate criterion does not match **do**
  $R_t = P_t \cup Q_t$
  $F = fast\_non\_dominated\_sort(R_t)$ {Assign individuals of R_t into different ranks.}
  set $P_{t+1} = \phi$ and $i = 1$
  while $|P_{t+1}| + |F_i| \leq N$ **do**
    $find\_crowding\_distance(F_t)$ {Find the crowding distances of individuals in F_t.}
    $P_{t+1} = P_{t+1} \cup F_t$
    i=i+1
  Sort(F_t)
  $P_{t+1} = P_{t+1} \cup F_t[1 : (N - \|P_{t+1}|)]$
  $Q_{t+1} = produce\_new\_population(P_{t+1})$ {Produce new individuals and evaluate them.}
  t = t + 1
set t = 0 { Reset t to zero}
define k { The size for initial ensemble}
define C_t {The provisional ensemble for comparison.}
E_t = random_select_individuals(k,R_t)
$R_t := R_t - E_t$ { Remove the selected individuals from R_t}
while terminate criterion does not match **do**
  C_t = random_select_individuals(k,R_t) {Randomly select k individuals from R_t.}
  if (E_t and C_t are significantly different) **then**
    $E_t = E_t \cup C_t$
    $R_t = R_t - C_t$
    k = k + k
  t = t + 1
return E_t

IV. EXPERIMENTS AND RESULTS

This section describes the experiment preparation and experiment results. In this study, a number of data mining techniques are applied to solve four financial fraud detection problems. The experiment preparation is described in Section IV-A. Section IV-B shows the parameter setting for the proposed method and briefly introduces several variants of the proposed method. Section IV-C presents the experiment results of all methods and discusses the results in detail.

A. Introduction to Experiment Preparation

In order to compare the performance of Grammar-based Multi-objective Genetic Programming with Statistical Selection Learning (GBMGP-SSL) and the other well-known data mining techniques, we apply Waikato Environment for Knowledge Analysis (WEKA) [63] in the experiments. Logistic Regression (LR), Neural Networks (NNs), Support Vector Machine (SVM), Bayesian Networks (BNs), Decision Trees (DTs), AdaBoost, Bagging, LogitBoost, variants of GBGP and variants of GBMGP (including the proposed method) are evaluated in the study.

1) Data Description:

Four financial fraud detection problems are considered. Two of them have been taken from the UCI machine learning repository [64] and the other two are real-life financial fraud problems. The description of datasets is shown in Table III.

TABLE III
Data Description

Dataset	Attributes	Instances	Classes	Class Ratio
Australian credit	14	690	2	307:383
Credit approval	15	690	2	307:383
U.S. CSF	41	68332	2	63:68269
CCSF	17	18373	2	855:17518

Australian credit and Credit approval are similar, but the latter has one more attribute, which may affect the results. However, they are often used together as benchmark problems in many data mining studies. Their class distributions are balanced.

For the U.S. corporate securities fraud (U.S. CSF) dataset, the original dataset has about 200 variables with duplicated and useless attributes, such as firm identity number and name. The dataset is extremely imbalanced, which may affect the results if models are learned directly from it. In general, the number of fraudulent firms is much smaller than the number of non-fraudulent firms. Therefore, it is better to maintain all fraudulent instances. Otherwise, it is difficult to learn the fraudulent information based on a small number of instances. If the fraudulent firms have too many missing values (e.g. more than 40% missing values) in some attributes, we remove those attributes directly. On the other hand, if the fraudulent firms have few missing values in some attributes, we replace them based on the data distributions of those attributes (e.g. take the mean of the variable as the value for the missing data). For non-fraudulent firms, we remove the instances with many missing values. For the attributes with few missing values, we also replace them based on the data distributions of those attributes.

The China corporate securities fraud (CCSF) dataset contains records of corporations with their firm, financial, governance and trade characteristics. The variables are selected on the basis of the related literature discussed in Section II-A2. Moreover, including more attributes may provide more interesting information of the fraudulent firms for learning. The original dataset has 21,396 instances with 24 attributes for all listed firms from 1998 to 2011. Each instance with more than 20 missing values in these 24 attributes is directly removed. Moreover, seven attributes about trade characteristics are removed since more than two-thirds of firms have not this trade information. The final dataset contains 18,373 records with 17 attributes. This dataset is also highly imbalanced with 4.7% fraudulent and 95.3% non-fraudulent examples.

2) Synthetic Minority Over-sampling Technique (SMOTE:

Imbalanced datasets cannot be directly used in some of the selected methods. Without prior consideration of the imbalance, the classifier(s) will always generate biased results for the majority class. Such classifiers are not useful, as their performance could be atrocious [65]. A number of approaches have been introduced to address imbalanced datasets, such as resampling techniques or pre-processing methods. The Synthetic Minority Over-sampling Technique (SMOTE) is a data pre-processing method, which can process data and generate synthetic examples by taking each minority class example along the line joining all of its k nearest neighbours. For example, if the number of minority class examples needed is triple (i.e. 300%), and the number of its nearest neighbours are limited to 5 (i.e. k=5). 3 of 5 nearest neighbors are selected as three directions and one synthetic example is generated along each direction. SMOTE is used in this study for a variety of reasons. First, SMOTE is very simple to implement in practice. Second, SMOTE has been shown empirically to perform well against random oversampling techniques in a lot of experiments [65], [66]. Third, the synthetic examples are generated in a less application-oriented manner.

3) K-folds Cross-Validation:

For a robust experiment, a ten-fold cross-validation mechanism is applied for each dataset. The ten-fold cross-validation splits the dataset into ten mutually exclusive and exhaustive folds. For each experiment, one fold is regarded as the testing dataset and the other nine folds are combined together as the training dataset. Figure 7 is a graphic illustration of ten-fold cross-validation.

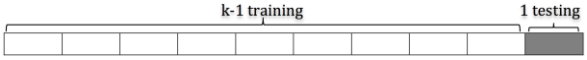

Fig 7: Ten-fold cross-validation example

All the learning approaches use the same training datasets to learn models and the same testing datasets are used to evaluate the performance of different models. The experiments are repeated for ten times until each fold is evaluated as a testing dataset. The average performance results are then reported. In addition, the two real-life datasets (i.e. U.S.CSF and CCSF) have imbalanced data distributions. When evaluating data mining methods on these two datasets, only the training datasets are pre-processed by SMOTE. In other words, the testing datasets maintain the original data distribution.

4) Model Evaluation Criteria:

As discussed in Section III-A, accurate rates are the most important criteria of a model when solving FFD problems. Each problem has two classes. The first class is regarded as *positive* (i.e. fraudulent) and the other is *negative* (i.e. non-fraudulent). Table IV shows the possible outcomes for binary classification.

TABLE IV
Contingency table with four outcomes of binary classification

	Classified as True	Classified as False
Actual is True	True Positive (TP)	False Negative (FN)
Actual is False	False Positive (FP)	True Negative (TN)

The accurate rate of positive class is called *true positive rate* (TPR), which is calculated by Equation (6).

$$TPR = TP / (TP + FN) \quad (6)$$

where TP is the number of positive examples that are correctly classified. TP + FN is the total number of positive examples including the number of correctly classified positive examples (i.e. TP) and the number of positive examples incorrectly classified as negative (i.e. FN). The accuracy rate for the negative class is called *true negative rate* (TNR), which is calculated by Equation (7).

$$TNR = TN / (TN + FP) \quad (7)$$

where TN is the number of negative examples that are correctly classified. TN + FP is the number of total negative examples including the number of correctly classified negative examples (i.e. TN) and the number of negative examples that are incorrectly classified as positive (i.e. FP). It is easy to observe the performance of each model for each class by using TPR and TNR as evaluation criteria.

B. Parameter Settings

Table V shows the parameter setting for the Grammar-based Multi-objectives Genetic Programming with Statistical Selection Learning (GBMGPSSL). In addition to GBMGP-SSL, several GBGP variants and GBMGP variants are developed for model comparisons. GBGP variants include GBGP(s, c), GBGP(s, c) with majority voting and GBGP(s, c) with weighted voting, where s and c indicate support and confidence respectively. In majority voting, the rules matching the testing case will make their own predictions about the class of the case and the final prediction is determined by the votes. On the other hand, each rule has a weight, which is the average of its support and confidence, for weighted voting and the final prediction for the testing case is determined by the weighted votes. GBMGP variants include GBMGP(s, c), GBMGP(s, c) with majority voting and GBMGP(s, c) with weighted voting. In GBGP variants, support and confidence are combined in a linear equation. On the other hand, support and confidence are the two objectives in GBMGP variants. GBGP(s) and GBGP(c) are special variants of GBGP that only consider support and confidence respectively in their fitness functions.

TABLE V
Parameters and values for the proposed method

GBMGP-SSL	
Parameter	**Value**
Population size	200
Max. no. of generation	500
Use elitism	no
Selection scheme	tournament
Tournament size	2
Crossover rate	0.8
Mutation rate	0.2
Ensemble method	statistical selection
Max. ensemble size	0.6

GBGP variants use elitism to select the best individual(s) of the current population for the next generation directly. The elitism operator always selects the individual with the highest fitness value for the next generation directly without using any genetic operators. GBMGP variants do not use elitism, since the non-dominated solutions in the current population are already considered in the evolutionary process automatically. Other experiments settings are the same as shown in Table V. Moreover, tournament selection is used. It randomly selects a number of solutions with tournament size *k*, and chooses the best (i.e. winner) for genetic operation (e.g. crossover or mutation). The default tournament size is 2. The last two parameters are only used in the proposed method for ensemble learning. It applies the proposed statistical selection with the maximum of 60 percents of the whole population. For example, if the population size is 100 and ensemble size setting is 0.6 (i.e. 60%), then at most 60 individuals will be selected to form an ensemble. On the other hand, majority voting and weighted voting used all evolved rules (i.e. ensemble size is 100%).

The parameter settings of other data mining methods are shown in Table VI and separated by a double line.

C. Results and Analysis

Table VII summarizes the average accuracies for each class on the four financial datasets. The name of each method is shown in the first column. Four datasets are evaluated by nine methods in this experiment. Each dataset has two classes: positive and negative, and the corresponding accuracies are indicated by TPR and TNR respectively, which are shown in the second row of Table VII. The Standard Deviation (S.D.) of each method is also given below the corresponding accuracy result. For example, Logistic Regression obtains 81% accurate rate in classifying positive class on the Australian dataset, and its S.D. is 5.8%.

TABLE VI
Parameter settings for the compared approaches

Method	Parameter	Value
Logistic Regression (LR)	Ridge	1.0E-8
	Max. iterations	-1
Neural Networks (NNs)	Learning rate	0.3
	Momentum Value	0.2
	No. hidden Layers	1
	Weight update	Back propagation
	Training epochs	500
	Random seed	0
Support Vector Machine (SVM)	Kernel function	Polykernel
	Complexity	1
	Tolerance rate	0.001
	Exponent value	1
Bayesian Networks (BNs)	Estimator	Simple estimator
	Search algorithm	Hill climbing
Decision Trees (DTs)	Min. number of nodes	2
	No pruning	False
	Number of folds	3
	Min. variance probability	0.001
AdaBoost	Classifier	Decision stump
	Number of iterations	20
	Seed	1
	Use resampling	False
	Weight threshold	100
Bagging	Classifier	REPTree
	Number of iterations	20
	Bag size percent	100
LogitBoost	Classifier	Decision stump
	Number of iterations	20
	Use resampling	False
	Seed	1
	Weight threshold	100
	Likelihood threshold	-1.798

For Australian credit and credit approval datasets, all the approaches are promising with regard to their TPRs and TNRs. For the two real-life datasets (U.S.CSF and CCSF), the performance values are not stable using different methods. Some methods such as Decision Trees and Bagging generate extremely biased results with very low TPRs and very high TNRs. Logistic Regression obtains about 41% in regard to classifying fraudulent firms in both real-life datasets. SVM, Bayesian Networks, AdaBoost and LogitBoost obtain about 50% in regard to classifying fraudulent firms for the U.S.CSF dataset only, and worse TPR for the CCSF dataset. The proposed method with minority prediction, which is located in the last row of Table VII can achieve better TPR results than all of the other techniques, but its TNR values for each dataset are relatively lower at the same time. According to the characteristics of financial datasets, especially for real life FFD problems, the detection of positive class (i.e. fraudulent) is much more important than the detection of negative class (i.e. non-fraudulent). For example, if a firm is fraudulent, and it is incorrectly classified as non-fraudulent, then the loss to interested people (e.g. shareholders) may be destructive. However, if a firm is non-fraudulent, and it is incorrectly classified as fraudulent, it may need to be investigated by the Securities and Exchange Commission (SEC) or the China Securities Regulatory Commission (CSRC) at relatively much lower cost (i.e. investigation fees) compared to the destructive consequence caused by fraud without any investigations. Therefore, it is more important to classify fraudulent firms correctly than non-fraudulent firms.

In order to have a more comprehensive comparison for the proposed method, a number of GBGP variants and GBMGP variants are developed and the corresponding comparison results are shown in Table VIII. The name of each method is located in the first column, and the meanings of notations are indicated in Table IX. For example, the first method is GBGP(s, c)$_a$, which is the original GBGP. The symbol "*a*" means that the first method used m**a**jority prediction. As another example, the last method is GBMGP(s, c, S)$_i$, which is the proposed method. It is a multi-objective GBGP (i.e. GBMGP), and the abbreviation "M" indicates that the system has the **m**ulti-objective component. Therefore, it uses support (i.e. s) and confidence (i.e. c) as the two objectives. The symbol "S" indicates that **s**tatistical selection learning is used as the ensemble learning method. The symbol "*i*" means that the last method uses m**i**nority prediction.

For the Australia credit and Credit approval datasets, the original GBGP can obtain about 85% accuracy for both TPRs and TNRs. Different ensemble learning techniques (i.e. majority voting and weighted voting) cannot improve the original GBGP, no matter whether majority prediction or minority prediction is used. In addition, variants with multi-objective (i.e. GBMGP) have slightly poorer performance in regard to TPRs and slightly better performance in regard to TNRs than the original GBGP. However, variants with multi-objective (i.e. GBMGP) and ensemble techniques perform similarly and even obtain better TPRs and TNRs than the original GBGP except for the proposed method. The proposed method obtains the highest TPRs and has slightly poorer performance in regard to TNRs.

For the U.S.CSF dataset, all methods using majority prediction have good performance in regard to TNRs. However the corresponding TPRs are very low, with only the GBMGP(s, c, S)$_a$ obtaining a result that is more than 50% for TPR. The TPRs are relatively improved by using minority prediction, but still less than 50%. The proposed method achieves 64%, which is the highest TPR value among all variants.

TABLE VII
Accuracies of data mining techniques and the proposed method

	Australia credit		Credit approval		U.S.CSF		CCSF		
Methods	TPR	TNR	TPR	TNR	TPR	TNR	TPR	TNR	w/t/l
LR	0.81^{++}	0.81	0.85^{++}	**0.84**$^{-}$	0.41^{++}	0.90^{--}	0.41^{++}	0.47^{++}	6/0/2
	0.058	0.063	0.046	0.042	0.222	0.013	0.045	0.017	
NNs	0.80^{++}	0.83	0.80^{++}	0.83	0.47^{++}	0.94^{--}	0.31^{++}	0.83^{--}	5/0/3
	0.089	0.064	0.047	0.048	0.191	0.014	0.067	0.053	
SVM	0.87	0.79^{++}	0.80^{++}	0.79	0.51^{+}	0.94^{--}	0.41^{++}	0.73^{--}	6/0/2
	0.027	0.069	0.043	0.051	0.183	0.006	0.037	0.019	
BNs	0.80^{++}	**0.88**	0.80^{++}	0.81	0.52^{+}	0.91^{--}	0.28^{++}	**0.94**$^{--}$	4/0/4
	0.059	0.017	0.046	0.029	0.185	0.008	0.022	0.006	
DTs	0.82^{++}	0.84	0.82^{++}	**0.84**	0.10^{++}	**1.00**$^{--}$	0.24^{++}	0.93^{--}	5/0/3
	0.079	0.105	0.078	0.103	0.071	0.001	0.029	0.009	
AdaBoost	0.81^{++}	0.84	0.77^{++}	0.81	0.51^{+}	0.88^{--}	0.47^{++}	0.71^{-}	5/0/3
	0.053	0.062	0.046	0.069	0.174	0.016	0.046	0.031	
Bagging	0.79^{++}	0.82	0.78^{++}	0.80	0.11^{++}	**1.00**$^{--}$	0.23^{++}	0.89^{--}	5/1/2
	0.068	0.067	0.057	0.058	0.076	0.001	0.025	0.008	
LogitBoost	0.81^{++}	0.82	0.83^{++}	0.82	0.50^{+}	0.90^{--}	0.44^{++}	0.80^{--}	5/0/3
	0.038	0.069	0.032	0.047	0.211	0.013	0.046	0.028	
GBMGP(s,c,S)$_i$	**0.89**	0.85	**0.89**	0.80	**0.64**	0.76	**0.66**	0.67	n/a
	0.063	0.069	0.058	0.055	0.118	0.133	0.074	0.069	

1. $^{++}$ Using paired t-test, the average accuracy is significantly worse than that of GBMGP(s,c,S)$_i$ at the 0.05 level.
2. $^{+}$ Using paired t-test, the average accuracy is significantly worse than that of GBMGP(s,c,S)$_i$ at the 0.1 level.
3. $^{-}$ Using paired t test, the average accuracy is significantly better than that of GBMGP(s,c,S)$_i$ at the 0.1 level.
4. $^{--}$ Using paired t-test, the average accuracy is significantly better than that of GBMGP(s,c,S)$_i$ at the 0.05 level.

For the CCSF dataset, GBMGP without using any ensemble learning methods cannot improve the results over the original GBGP. The GBGP with majority voting even produces poorer TPR results. However, compared to the original GBGP, the TPR of GBGP with minority prediction has about 22.9% improvements. Except for the proposed method, the GBGP with majority voting and minority prediction obtains the second highest TPR, but the corresponding TNR is greatly reduced. The minority prediction performs well in this dataset, especially for GBGP(s,c,M)$_i$. Finally, the proposed method produces the highest TPR and relatively higher TNR compared to GBGP(s, c, M)*i*.

The following empirical and statistical tests focus on the comparison between GBMGP-SSL with minority prediction and the other approaches.

1) Empirical Analysis:

Each problem has two results for TPR and TNR respectively. We set each one as a competition, and therefore eight competitions are performed for the four datasets. The empirical *w/t/l* (i.e. win, tie and lose) test results are given in the last column of Table VII and Table VIII, where *w* means that GBMGP-SSL with minority prediction outperforms the compared approach, *t* means that GBMGP-SSL with minority prediction has the same results, and *l* means that GBMGP-SSL with minority prediction is worse than the compared approach.

In Table VII, compared with Bayesian Networks (BNs), the proposed method wins all competitions in regard to TPR, but also loses all of them with regard to TNR. BNs can be regarded as a generic method for solving FFD problems. Although it obtains the highest TNR for Australia and the second highest TPR for U.S.CSF, it also gets bad TPR results in the CCSF dataset. In this study, the real-life datasets are more important than the benchmark datasets. Compared with Logistic Regression (LR), the proposed method wins six competitions and loses two, which are TNRs from credit approval and U.S.CSF. Especially for U.S.CSF, LR has a biased result for TNR. However, logistic regression is still a competitive method compared with other approaches. Compared with Neural Networks (NNs), Support Vector Machine (SVM) and Decision Trees (DTs), the proposed method respectively wins 5, 6 and 5 competitions. On the other hand, it respectively loses in 3, 2 and 3 competitions, which are also related to TNRs. SVM has similar performance to that of BNs, but DTs has extremely biased results with regard TNRs. Therefore, DTs may not be an appropriate method for imbalanced financial datasets. Compared with AdaBoost, Bagging and LogitBoost, the proposed method respectively wins 5, 5 and 5 competitions, and respectively ties in 0, 1 and 0 competitions. Therefore, the proposed method is able to outperform other ensemble learning techniques even in the TNR

competitions. Bagging generates extremely biased TNR results on the two real-life financial fraud datasets, and thus it may not be an appropriate method for imbalanced financial datasets. In addition, comparing the ensemble learning techniques with the LR for U.S.CSF and CCSF datasets, the overall TPR results of ensemble learning techniques (except for Bagging) improve.

TABLE VIII
Classification accuracies of the proposed method and its variants

	Australia credit		Credit approval		U.S.CSF		CCSF		
Methods	TPR	TNR	TPR	TNR	TPR	TNR	TPR	TNR	w/t/l
GBGP(s,c)$_a$	0.84^{+}	0.86	0.84^{++}	0.85^{--}	0.33^{++}	0.84^{--}	0.48^{++}	0.78^{--}	4/0/4
	0.063	0.095	0.027	0.045	0.201	0.049	0.059	0.025	
GBGP(c)$_a$	0.68^{++}	0.79^{++}	0.66^{++}	0.75^{+}	0.29^{++}	0.83^{-}	0.33^{++}	0.65	7/0/1
	0.049	0.024	0.045	0.060	0.046	0.054	0.062	0.045	
GBGP(s)$_a$	0.09^{++}	0.97^{--}	0.09^{++}	0.97^{--}	0.02^{++}	0.93^{--}	0.07^{++}	0.94^{--}	4/0/4
	0.048	0.027	0.015	0.018	0.009	0.031	0.033	0.034	
GBGP(s,c, M)$_a$	0.82^{++}	0.79^{++}	0.82^{++}	0.79	0.39^{++}	0.85^{--}	0.25^{++}	$\mathbf{0.92}^{--}$	6/0/2
	0.034	0.075	0.032	0.031	0.254	0.021	0.034	0.021	
GBGP(s,c, W)$_a$	0.82^{++}	0.80^{+}	0.84^{++}	0.79	0.30^{++}	$\mathbf{0.89}^{--}$	0.54^{++}	0.67	6/1/1
	0.046	0.053	0.031	0.041	0.120	0.046	0.055	0.051	
GBMGP(s,c)$_a$	0.80^{++}	$\mathbf{0.91}^{--}$	0.80^{++}	0.84	0.44^{++}	0.86^{--}	0.48^{++}	0.73^{--}	4/0/4
	0.054	0.027	0.074	0.066	0.229	0.034	0.058	0.054	
GBMGP (s,c,W)$_a$	0.85	0.90^{--}	0.85^{+}	$\mathbf{0.89}^{--}$	0.32^{++}	0.83^{-}	0.38^{++}	0.88^{--}	4/0/4
	0.061	0.027	0.053	0.036	0.165	0.046	0.108	0.052	
GBMGP (s,c, M)$_a$	0.84^{++}	$\mathbf{0.91}^{--}$	0.84^{++}	$\mathbf{0.89}^{--}$	0.45^{++}	0.83	0.48^{++}	0.87^{--}	4/0/4
	0.040	0.047	0.034	0.049	0.235	0.084	0.127	0.061	
GBMGP(s,c,S)$_a$	0.82^{++}	0.90	0.86	0.87^{--}	0.54^{+}	$\mathbf{0.89}^{--}$	0.53^{++}	0.81^{--}	4/0/4
	0.081	0.079	0.056	0.049	0.122	0.089	0.081	0.084	
GBGP(s,c)$_i$	0.82^{++}	0.78^{++}	0.83^{++}	0.80	0.43^{++}	0.81	0.59^{+}	0.59^{++}	6/1/1
	0.043	0.072	0.038	0.059	0.179	0.047	0.092	0.096	
GBGP(c)$_i$	0.75^{++}	0.69^{++}	0.73^{++}	0.68^{++}	0.34^{++}	0.58^{++}	0.43^{++}	0.49^{++}	8/0/0
	0.061	0.050	0.075	0.058	0.155	0.137	0.050	0.047	
GBGP(s)$_i$	0.13^{++}	0.99^{--}	0.14^{++}	0.97^{--}	0.08^{++}	0.97^{--}	0.10^{++}	0.96^{--}	4/0/4
	0.031	0.018	0.035	0.027	0.036	0.027	0.047	0.038	
GBGP(s,c, M)$_i$	0.84^{++}	0.78^{++}	0.85^{++}	0.79	0.47^{++}	0.86^{--}	0.61^{+}	0.60^{++}	7/0/1
	0.040	0.067	0.025	0.042	0.189	0.033	0.058	0.057	
GBGP(s,c, W)$_i$	0.85^{++}	0.79^{++}	0.86	0.79	0.44^{++}	0.76	0.60^{++}	0.58^{++}	7/1/0
	0.038	0.036	0.037	0.062	0.116	0.065	0.062	0.040	
GBMGP(s,c)$_i$	0.82^{++}	0.88	0.84^{++}	0.86^{--}	0.45^{++}	0.73	0.48^{++}	0.58^{++}	6/0/2
	0.037	0.070	0.045	0.048	0.258	0.143	0.052	0.055	
GBMGP (s,c,W)$_i$	0.86	0.89	0.85^{+}	0.87^{--}	0.35^{++}	0.81	0.55^{++}	0.68	4/0/4
	0.067	0.050	0.055	0.036	0.113	0.063	0.053	0.031	
GBMGP (s,c, M)$_i$	0.85	0.86	0.82^{++}	0.84^{-}	0.44^{++}	0.75	0.59^{++}	0.64	6/0/2
	0.064	0.087	0.047	0.022	0.186	0.154	0.042	0.075	
GBMGP(s,c,S)$_i$	**0.89**	0.85	**0.89**	0.80	**0.64**	0.76	**0.66**	0.67	n/a
	0.063	0.069	0.058	0.055	0.118	0.133	0.074	0.069	

1. $^{++}$ Using paired t-test, the average accuracy is significantly worse than that of GBMGP(s,c,S)$_i$ at the 0.05 level.
2. $^{+}$ Using paired t-test, the average accuracy is significantly worse than that of GBMGP(s,c,S)$_i$ at the 0.1 level.
3. $^{-}$ Using paired t-test, the average accuracy is significantly better than that of GBMGP(s,c,S)$_i$ at the 0.1 level.
4. $^{--}$ Using paired t-test, the average accuracy is significantly better than that of GBMGP(s,c,S)$_i$ at the 0.05 level.

TABLE IX
Abbreviations of all the approaches

Abbreviation	Description
s	Objective: support
c	Objective: confidence
W	Ensemble: Weighted voting
M	Ensemble: Majority voting
S	Ensemble: Statistical selection
a	Majority prediction
i	Minority prediction

According to Table VIII, compared with GBGP(s,c)$_a$, GBMGP(s,c,W)$_i$, GBMGP(s,c)$_a$, GBMGP(s,c,W)$_a$, GBMGP(s,c,M)$_a$ and GBMGP(s,c,S)$_a$, the proposed method wins all of the competitions for TPR, but also loses all of them for TNR. Compared with GBGP(s,c,M)$_a$, GBGP(s,c)$_i$, GBGP(s,c,W)$_a$, GBGP(s,c)$_i$, GBGP(s,c,M)$_i$ and GBGP(s,c,W)$_i$, the proposed method respectively wins 6, 6, 6, 6, 7 and 7 competitions, and respectively ties in 0, 0, 1, 1, 0 and 1. This indicates that the proposed method outperforms other GBGP variants no matter whether majority or minority prediction is applied. Moreover, it also indicates that the multi-objective and the new statistical selection learning technique together can improve the results for most TPRs and TNRs.

Finally, compared with GBMGP(s,c,M)$_i$ and GBMGP(s,c)$_i$, the proposed method wins 6 and loses in 2 competitions. The two lost competitions are for TNRs from the Australian credit and credit approval datasets. This indicates that the GBMGP(s,c,M)$_i$ and

GBMGP(s,c)$_i$ with minority prediction maybe more suitable for balanced datasets. Comparing GBMGP(s,c,M)$_i$ and GBMGP(s,c)$_i$, it can be found that majority voting can improve the TPR and TNR results for the CCSF dataset.

In addition, relative improvement (RAI) is applied to evaluate the approaches [59]. RAI is calculated by using Equation (8).

$$p = \sum \frac{a_i - b_i}{b_i} \tag{8}$$

where a_i denotes the accuracy of the GBMGP-SSL with minority prediction in the i^{th} dataset and b_i refers to the accuracy of the approach being compared with. In Table X, RAI (TPR) indicates the relative improvements for TPR and RAI (TNR) indicates the relative improvements for TNR. According to Table X, the proposed method outperform LR, GBGP variants with minority prediction, and GBMGP(s,c)$_i$, because RAI(TPR) and RAI(TNR) are larger than zero. For most of the other techniques, the proposed method reduces the TNR results slightly while achieves significant improvements on the TPR results, especially for Bagging, DTs, GBGP(s)$_a$ and GBGP(s)$_i$.

TABLE X
RAI test result

	RAI (TPR)	RAI (TNR)
Logistic Regression (LR)	130%	25%
Neural Networks (NNs)	170%	-39%
Support Vector Machine (SVM)	98%	-19%
Bayesian Networks (BNs)	186%	-51%
Decision Trees (DTs)	720%	-56%
AdaBoost	91%	-21%
Bagging	712%	-46%
LogitBoost	94%	-31%
GBGP(s,c)$_a$	141%	-31%
GBGP(c)$_a$	292%	6%
GBGP(s)$_a$	5,358%	-78%
GBGP(s,c, M)$_a$	243%	-30%
GBGP(s,c, W)$_a$	152%	-7%
GBMGP(s,c)$_a$	105%	-32%
GBMGP (s,c,W)$_a$	182%	-50%
GBMGP (s,c, M)$_a$	93%	-49%
GBMGP(s,c,S)$_a$	48%	-47%
GBGP(s,c)$_i$	73%	16%
GBGP(c)$_i$	184%	108%
GBGP(s)$_i$	2,397%	-84%
GBGP(s,c, M)$_i$	53%	9%
GBGP(s,c, W)$_i$	64%	23%
GBMGP(s,c)$_i$	96%	6%
GBMGP (s,c,W)$_i$	111%	-22%
GBMGP (s,c, M)$_i$	69%	-1%
GBMGP(s,c,S)$_i$	-	-

2) Statistical Analysis:

Pairwise t-test is applied to demonstrate the statistical significance of the experiments. The performance of the proposed method and other approaches is compared to calculate statistical significance. The results of the t-test are shown in Table VII and Table VIII, which use the symbol "++" to indicate that the proposed method is significantly better than the compared method at the 5% level and apply the symbol "+" to represent that the proposed method is significantly better than the compared method at the 10% level. On the other hand, the symbols "−−" and "−" are used if the proposed method is significantly worse than the compared method at the 5% level and 10% level respectively. For example in Table VII, compared with LR, the proposed method significantly outperforms it in 5 of the 8 metrics at the 5% level. However, the proposed method is significantly worse than LR for TNRs on credit approval and U.S.CSF at the 10% level and 5% level, respectively. In addition, the proposed method is significantly better than NNs in 4 of the 8 metrics at the 5% level. Compared with SVM, it is significantly superior in 3 of the 8 metrics at the 5% level, and 1 metric at the 10% level. Compared with BNs, it is significantly superior in 3 of the 8 metrics at the 5% level, and 1 metric at the 10% level. It is also significantly superior to DTs in 4 metrics at the 5% level. Moreover, it also outperforms AdaBoost and LogitBoost in 3 of the 8 metrics at the 5% level, and 1 metric at the 10% level. It outperforms Bagging in 4 of the 8 metrics at the 5% level.

For the benchmark problems in Table VII, the proposed method significantly outperformed all the data mining techniques for TPRs, except for SVM on Australia credit. Moreover, it also significantly outperforms SVR for TNRs on Australia credit at the 5% level, but is significantly worse than the LR for TNRs on Credit approval at the 10% level.

For the U.S.CSF dataset, the proposed method significantly outperforms LR, NNs, DTs and Bagging for TPRs at the 5% level, and it also significantly outperforms SVM, BNs, AdaBoost and LogitBoost for TPRs at the 10% level. However, the TNRs of using the proposed method is significantly worse than all the data mining methods at the 10% level.

For the CCSF dataset, the proposed method significantly outperforms all the data mining methods for TPRs at the 5% level, but is also significantly worse than all the data mining methods at the 10% level, except for the LR. It is easy to obtain very good results on the majority (i.e. negative) class by applying the compared approaches. As discussed before, the detection of fraudulent firms is much more important than the correct classification of non-fraudulent firms. The proposed method seems to reduce by a few percent of the accurate rate of classifying non-fraudulent firms, but it significantly increases the performance in identifying fraudulent firms. Thus the proposed method is promising for FFD problems.

V. CONCLUSION

A. Major Findings

Financial fraud has become an increasingly serious problem in economics, finance and management. Financial fraud detection (FFD) is vital for the prevention of the destructive consequences of

financial fraud. Data mining plays a significant role in solving FFD problems. In this study, we conduct a comprehensive comparison of data mining techniques and suggest a new approach to identify fraudulent cases from four financial datasets. The applied data mining techniques include Logistic Regression (LR), Neural Networks (NNs), Support Vector Machine (SVM), Bayesian Networks (BNs), Decision Trees (DTs), AdaBoost, Bagging and LogitBoost. We can conclude several findings from the experiment results.

1) The performance of the existing data mining methods on the given FFD datasets:

All the applied data mining techniques perform well on the benchmark datasets. For the Australia credit dataset, SVM can produce better classification results for positive class (i.e. fraudulent) than other methods. It also gives very good performance in classifying the negative examples (i.e. non-fraudulent), but BNs achieves the best performance in this aspect. For the Credit approval dataset, LR can generate the highest results for both classes. Moreover, DTs also produces the highest accuracy in classifying the negative examples, but the corresponding standard derivation is larger than that of LR. Ensemble learning techniques cannot produce outstanding results for the benchmark datasets.

Real-life problems are more challenging. All the methods perform poorly in classifying fraudulent firms, but well for non-fraudulent firms classification. For the U.S.CSF dataset, all methods obtain poor results with very high standard derivations for fraudulent firms classification. DTs and Bagging obtain the lowest accuracies in classifying the fraudulent firms, but their accuracies of the negative class classification are the highest. This implies that these methods cannot be used for this dataset, since the results are extremely biased towards the majority class (i.e. non-fraudulent firms). For the CCSF dataset, the situation is similar to that for the U.S.CSF dataset, all the methods can not perform well for the fraudulent class. However, the corresponding standard derivations are much lower compared to the situation in the U.S.CSF dataset. AdaBoost and LogitBoost produce relatively better results than the other methods on the CCSF dataset.

2) The performance of GBGP variants and GBMGP variants for the given FFD datasets:

In order to evaluate the features of the proposed method, such as Multi-objective optimization and ensemble techniques, we develop several GBGP variants and GBMGP variants and compare their performance with the original GBGP. For the benchmark datasets, all GBGP variants and GBMGP variants using minority prediction cannot improve significantly the classification results for both classes. But GBMGP variants using majority prediction and different ensemble methods (i.e. weighted voting and majority voting) improve the average accuracy of the negative (i.e. non-fraudulent) class classification. For the U.S.CSF and CCSF datasets, many of the GBMGP variants produce better results than the original GBGP. Minority prediction improves all GBGP variants and GBMGP variants in regard to classifying the positive (i.e. fraudulent) examples. Ensemble methods in GBGP variants do not improve the results significantly. But ensemble methods with GBMGP variants to varying degrees produce better results for the positive class.

Moreover, all GBGP variants and GBMGP variants produce competitive results compared with the data mining methods.

3) The proposed method and comparison with all other methods:

The proposed method produces significantly better results than most of the other methods. Especially for fraudulent classification (i.e. positive class), the proposed method obtains the highest True Positive Rates (TPRs) for the given FFD datasets. However, the non-fraudulent class (i.e. negative class) classification accuracies decline slightly. The proposed method is promising as it improves the performance of the fraudulent detection significantly, and it is much more important to detect fraudulent cases than non-fraudulent cases.

B. Contributions and Implications

There are two major contributions of this study.

Firstly, we have performed a number of comprehensive comparisons between different data mining approaches in solving the FFD problem. These approaches include Logistic Regression, Neural Networks, Support Vector Machine, Bayesian Networks, Decision Trees, AdaBoost, Bagging and LogitBoost. Moreover, we have also developed a number of GBGP variants and GBMGP variants with different ensemble methods for comparison.

Secondly, we have proposed a new method called Grammar-based Multi-objective Genetic Programming with Statistical Selection Learning (GBMGP-SSL) that can take advantages of Multi-Objective Evolutionary Algorithms and ensemble learning. A new ensemble technique called Statistical Selection Learning (SSL) has been developed. SSL can outperform majority voting and weighted voting in classifying fraudulent firms from the two real-life FFD datasets.

By selecting a number of good classification rules, GBMGP-SSL can detect fraudulent instances. The performance of each individual classification rule is optimized considering its support and confidence values simultaneously. The comprehensive experiment results show that GBMGP-SSL performs well on the given FFD problems.

The major findings from this study may have important implications for the Securities and Exchange Commission (SEC) and China Securities

Regulatory Commission (CSRC) to facilitate their work. Researchers can utilize the comparison results between GBMGP-SSL and existing data mining methods for the four FFD problems as references for doing data mining research in financial fraud detection.

The application of classification rules in FFD provides understandable results for users, even if they are not experts in the relevant areas. The combination of GBGP, Multi-Objective Evolutionary Algorithms, and SSL also provides a novel and powerful approach for data mining. The stakeholders of SEC and CSRC may consider the possibility of applying GBMGP-SSL and other data mining methods for detecting financial frauds.

C. Directions for Future Research

It will be interesting to incorporate other objectives such as risk and return into the proposed method and evaluate it on different real-life financial datasets to determine whether some useful and interesting rules can be discovered.

The proposed method can be used to solve other business problems such as direct marketing problems [67]. In direct marketing, the evaluation of a method is usually based on response rate and total profit. Response rate is the ratio of the number of respondents to the total number of customers in the dataset. Total profit is the sum of the profits generated from all respondents. A high value of response rate may not produce high total profit. Therefore, it is necessary to find respondents who can contribute high profits. In this problem, the minority class is the high-profit customers and the majority class contains the low-profit customers and the non-respondents. The learned rules can identify the high-profit customers, low-profit customers and non-respondents if the objectives of the proposed method are changed to response rate and total profit.

The proposed method applies NSGA-II as the multi-objective optimization algorithm. We will study the effect of using other multi-objective algorithms on the performance of the proposed method.

ACKNOWLEDGMENT

This research is supported by the LEO Dr David P. Chan Institute of Data Science and the General Research Fund LU310111 from the Research Grant Council of the Hong Kong Special Administrative Region.

REFERENCES

[1] E. W. T. Ngai, Y. Hu, Y. H. Wong, Y. Chen, and X. Sun, "The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature," Decision Support Systems, vol. 50, no. 3, pp. 559–569, 2011.

[2] M. Syeda, Y.-Q. Zhang, and Y. Pan, "Parallel granular neural networks for fast credit card fraud detection," in Proceedings of the 2002 IEEE International Conference on Fuzzy Systems, vol. 1. IEEE, 2002, pp. 572–577.

[3] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, " Data mining for credit card fraud: A comparative study," Decision Support Systems, vol. 50, no. 3, pp. 602–613, 2011.

[4] P. Ravisankar, V. Ravi, G. Raghava Rao, and I. Bose, "Detection of financial statement fraud and feature selection using data mining techniques," Decision Support Systems, vol. 50, no. 2, pp. 491–500, 2011.

[5] D. Cumming, W. Hou, and E. Lee, "The role of financial analysts in deterring corporate fraud in China," SSRN Electronic Journal, 2011.

[6] Y. Kou, C.-T. Lu, S. Sirwongwattana, and Y.-P. Huang, "Survey of fraud detection techniques," in Proceedings of 2004 IEEE international conference on networking, sensing and control, vol. 2. IEEE, 2004, pp. 749–754.

[7] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011.

[8] I. H. Witten, E. Frank, and M. A. Hall, Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann Publishers Inc., 2011.

[9] I. Bose and R. K. Mahapatra, "Business data mining: a machine learning perspective," Information & management, vol. 39, no. 3, pp. 211–225, 2001.

[10] E. Turban, R. Sharda, D. Delen, J. Aronson, T. Liang, and D. King, Decision Support and Business Intelligence Systems, 9th ed. Pearson Prentice Hall, 2010.

[11] C. Phua, V. Lee, K. Smith, and R. Gayler, "A comprehensive survey of data mining-based fraud detection research," arXiv preprint arXiv:1009.6119, 2010.

[12] G. Chen, M. Firth, D. N. Gao, and O. M. Rui, "Ownership structure, corporate governance, and fraud: Evidence from china," Journal of Corporate Finance, vol. 12, no. 3, pp. 424–448, 2006.

[13] A. Agrawal, S. Chadha, M. Billett, R. Boylan, M. Chen, J. Engl, J. Jaffe, S. Krishnaswami, S. Lee, F. L. de silanes, N. R. Prabhala, Y. Qian, D. Reeb, R. Romano, P. K. Sen, and M. Stone, "Corporate governance and accounting scandals," Journal of law and economics, vol. 48, no. 2, pp. 371–406, 2005.

[14] B. E. Hermalin and M. S. Weisbach, "Information disclosure and corporate governance," The Journal of Finance, vol. 67, no. 1, pp. 195–233, 2012.

[15] T. Y. Wang, A. Winton, and X. Yu, "Corporate fraud and business conditions: Evidence from IPOs," The Journal of Finance, vol. 65, no. 6, pp. 2255–2292, 2010.

[16] G. Chen, M. Firth, D. N. Gao, and O. M. Rui, "Is China's securities regulatory agency a toothless tiger? evidence from enforcement actions," Journal of Accounting and Public Policy, vol. 24, no. 6, pp. 451–488, 2005.

[17] E. Kirkos, C. Spathis, and Y. Manolopoulos, "Data mining techniques for the detection of fraudulent financial statements," Expert Systems with Applications, vol. 32, no. 4, pp. 995–1003, 2007.

[18] S. Kotsiantis, E. Koumanakos, D. Tzelepis, and V. Tampakas, " Forecasting fraudulent financial statements using data mining," International Journal of Computational Intelligence, vol. 3, no. 2, pp. 104–110, 2006.

[19] J. W. Lin, M. I. Hwang, and J. D. Becker, "A fuzzy neural network for assessing the risk of fraudulent financial reporting," Managerial Auditing Journal, vol. 18, no. 8, pp. 657–665, 2003.

[20] I. Yeh, C.-h. Lien et al., "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients," Expert Systems with Applications, vol. 36, no. 2, pp. 2473–2480, 2009.

[21] S. Haykin, Neural networks: a comprehensive foundation. Prentice Hall PTR, 1994.

[22] L. Breiman, "Bagging predictors," Machine learning, vol. 24, no. 2, pp. 123–140, 1996.

[23] R. E. Schapire, "The strength of weak learnability," Machine learning, vol. 5, no. 2, pp. 197–227, 1990.

[24] G. Cui, M. L. Wong, and H.-K. Lui, "Machine learning for direct marketing response models: Bayesian networks with

evolutionary programming," Management Science, vol. 52, no. 4, pp. 597–612, 2006.
[25] C. C. Coello, G. B. Lamont, and D. A. Van Veldhuizen, Evolutionary algorithms for solving multi-objective problems. Springer, 2007.
[26] A. Ponsich, A. L. Jaimes, and C. A. C. Coello, "A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications," IEEE Transactions on Evolutionary Computation, vol. 17, no. 3, pp. 321–344, 2013.
[27] H. Zhao, "A multi-objective genetic programming approach to developing pareto optimal decision trees," Decision Support Systems, vol. 43, no. 3, pp. 809–826, 2007.
[28] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182–197, 2002.
[29] M. L. Wong and K. S. Leung, Data mining using grammar based genetic programming and applications. Kluwer Academic Publisher, 2000.
[30] M. L. Wong and K. S. Leung, "Evolutionary program induction directed by logic grammars," Evolutionary Computation, vol. 5, no. 2, pp. 143–180, 1997.
[31] M. L. Wong, "A flexible knowledge discovery system using genetic programming and logic grammars," Decision Support Systems, vol. 31, no. 4, pp. 405–428, 2001.
[32] A. Konak, D. W. Coit, and A. E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial," Reliability Engineering & System Safety, vol. 91, no. 9, pp. 992–1007, 2006.
[33] H. Eskandari and C. D. Geiger, "A fast pareto genetic algorithm approach for solving expensive multiobjective optimization problems," Journal of Heuristics, vol. 14, no. 3, pp. 203–241, 2008.
[34] J.-H. Wang, Y.-L. Liao, T.-m. Tsai, and G. Hung, "Technology based financial frauds in Taiwan: Issues and approaches," in Proceedings of the 2006 IEEE International Conference on Systems, Man and Cybernetics, 2006, pp. 1120–1124.
[35] R. J. Bolton and D. J. Hand, "Statistical fraud detection: A review," Statistical Science, vol. 17, no. 3, pp. 235–255, 2002.
[36] A. Srivastava, A. Kundu, S. Sural, and A. K. Majumdar, "Credit card fraud detection using hidden markov model," IEEE Transactions on Dependable and Secure Computing, vol. 5, no. 1, pp. 37–48, 2008.
[37] G. Geis and P. Jesilow, White-collar crime. Sage Periodicals Press, 1993.
[38] FBI, "Financial crimes report to the public 2007," Department of Justice, United States, Tech. Rep., 2007.
[39] S. L. Gillan, "Recent developments in corporate governance: An overview," Journal of corporate finance, vol. 12, no. 3, pp. 381–402, 2006.
[40] F. Yu and X. Yu, "Corporate lobbying and fraud detection," Journal of Financial and Quantitative Analysis, vol. 46, no. 06, pp. 1865–1891, 2012.
[41] A. Dyck, A. Morse, and L. Zingales, "Who blows the whistle on corporate fraud?" The Journal of Finance, vol. 65, no. 6, pp. 2213–2253, 2010.
[42] D. B. Farber, "Restoring trust after fraud: Does corporate governance matter?" The Accounting Review, vol. 80, no. 2, pp. 539–561, 2005.
[43] J. D. Cox, R. S. Thomas, and D. Kiku, "SEC enforcement heuristics: An empirical inquiry," Duke Law Journal, vol. 53, pp. 737–779, 2003.
[44] R. Brause, T. Langsdorf, and M. Hepp, "Neural data mining for credit card fraud detection," in Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence. IEEE, 1999, pp. 103–106.
[45] A. Shen, R. Tong, and Y. Deng, "Application of classification models on credit card fraud detection," in Proceedings of the 2007 International Conference on Service Systems and Service Management. IEEE, 2007, pp. 1–4.
[46] P. K. Chan and S. J. Stolfo, "Toward scalable learning with non-uniform class and cost distributions: A case study in credit card fraud detection," in Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining. AAAI Press, 1998, pp. 164–168.
[47] A. Dal Pozzolo, O. Caelen, Y.-A. Le Borgne, S. Waterschoot, and G. Bontempi, "Learned lessons in credit card fraud detection from a practitioner perspective," Expert Systems with Applications, vol. 41, no. 10, pp. 4915–4928, 2014.
[48] D. Sánchez, M. Vila, L. Cerda, and J.-M. Serrano, "Association rules applied to credit card fraud detection," Expert Systems with Applications, vol. 36, no. 2, pp. 3630–3640, 2009.
[49] J. Yuan, C. Yuan, and X. Deng, "The effects of manager compensation and market competition on financial fraud in public companies: An empirical study in China." International Journal of Management, vol. 25, no. 2, 2008.
[50] C. T. Spathis, "Detecting false financial statements using published data: some evidence from Greece," Managerial Auditing Journal, vol. 17, no. 4, pp. 179–191, 2002.
[51] W. Zhou and G. Kapoor, "Detecting evolutionary financial statement fraud," Decision Support Systems, vol. 50, no. 3, pp. 570–575, 2011.
[52] B. Bai, J. Yen, and X. Yang, "False financial statements: characteristics of china's listed companies and cart detecting approach," International journal of information technology & decision making, vol. 7, no. 2, pp. 339–359, 2008.
[53] J. H. Holland, Adaptation in Natural and Artificial Systems. Ann Arbor, MI, USA: The University of Michigan Press, 1975.
[54] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning. Reading, MA, USA: Addison-Wesley, 1989.
[55] J. R. Koza, Genetic programming: on the programming of computers by means of natural selection. MIT press, 1992.
[56] R. Poli, W. Langdon, and N. F. McPhee, A Field Guide to Genetic Programming. LuLu Enterprises, 2008.
[57] M. A. Keane, M. J. Streeter, W. Mydlowec, G. Lanza, and J. Yu, Genetic programming IV: Routine human-competitive machine intelligence. Springer, 2006, vol. 5.
[58] T. G. Dietterich, "Ensemble methods in machine learning," in Proceedings of the First International Workshop on Multiple classifier systems. Springer, 2000, pp. 1–15.
[59] Y. Chen, M.-L. Wong, and H. Li, "Applying ant colony optimization to configuring stacking ensembles for data mining," Expert Systems with Applications, vol. 41, no. 6, pp. 2688–2702, 2014.
[60] D. H. Wolpert, "Stacked generalization," Neural networks, vol. 5, no. 2, pp. 241–259, 1992.
[61] J. E. Hopcroft, Introduction to Automata Theory, Languages, and Computation, 3/E. Pearson Education India, 2008.
[62] R. Agrawal, T. Imieli´nski, and A. Swami, "Mining association rules between sets of items in large databases," in ACM SIGMOD Record, vol. 22, no. 2. ACM, 1993, pp. 207–216
[63] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," ACM SIGKDD explorations newsletter, vol. 11, no. 1, pp. 10–18, 2009.
[64] A. Asuncion and D. Newman, UCI machine learning repository, 2007.
[65] A. Liu, J. Ghosh, and C. E. Martin, "Generative oversampling for mining imbalanced datasets," in Proceedings of the 2007 International Conference on Data Mining, 2007, pp. 66–72.
[66] N. V. Chawla, K.W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," Journal of Artificial Intelligent Research, vol. 16, no. 1, pp. 321–357, 2012.
[67] G. Cui, M. L. Wong, and X. Wan, "Cost-sensitive learning via priority sampling to improve the return on marketing and CRM investment," Journal of Management Information Systems, vol. 29, no. 1, pp. 341–373, 2012.